Luglio 1984 - N. 1 - Lire 3.000 Anno 1 - Sped. Abb. post. gr. III/70 - Distr. Me Pe

COMMODORE

MENSILE PER UTENTI DI VIC 20 & COMMODORE 64

FLUSSI RELATIVI

•

COME PROTEGGERSI DAGLI SPROTETTORI

•

NUMERI PRIMI

•

RTTY

•

UN LIST PIÚ FACILE

•

DUM 1541

•

STARTER KIT

NON É UN EDITORIALE	*di Gloriano Rossi*	**5**
FLUSSI RELATIVI	*di Gloriano Rossi*	**6**
STARTER KIT	*di Giancarlo de Cobelli*	**13**
UN LIST PIÚ FACILE	*di Tullio Spezia*	**20**
RTTY MANAGEMENT SYSTEM	*di Ambrogio Mainini e Paolo Denaro*	**21**
COME PROTEGGERSI DAGLI SPROTETTORI	*di Ernesto Sidotti*	**29**
NUMERI PRIMI	*di Eugenio Coppari*	**31**
DUM 1541	*di Miki*	**41**

DIRETTORE RESPONSABILE
Agostina Ronchetti

REDATTORE CAPO
Gloriano Rossi

SEGRETARIA DI REDAZIONE
Maura Ceccaroli

GRAFICA e IMPAGINAZIONE
Renato Caruso
Francesco Amatori

FOTO
Franco Vignati

DIFFUSIONE E ABBONAMENTI
Marina

DIREZIONE, REDAZIONE
V.le Famagosta, 75
20142 Milano - Tel. 02/8466675
Autorizzazione del Tribunale di Milano
n. 103 del 25/2/84

STAMPA
Lito 3 (Cologno M.)

Concessionario esclusivo
per la diffusione - MEPE spa
Via G. Carcano, 32 - Milano

Spedizione in abbonamento
postale - Gruppo III/70

Prezzo della rivista L. 3.000
Numero arretrato L. 6.000

Abbonamento annuo L. 25.000
I versamenti vanno indirizzati,
a: Commodore C.C.
V.le Famagosta, 75 - 20145 Milano,
mediante emissione di assegno
bancario utilizzando il c/c postale
n.ro 31532203

Per i cambi di indirizzo, indicare,
oltre naturalmente al nuovo, anche
l'indirizzo precedente, ed allegare
alla comunicazione l'importo
di L. 500 anche in francobolli.

COMMODORE

NON È UN EDITORIALE

Tanto non lo leggerebbe nessuno! Infatti son fin troppo sicuro che la maggior parte dei lettori, se non la totalità, non legge mai gli editoriali delle riviste.

Ma allora perchè questa pagina?

La ragione è presto detta.

Quando girerete pagina ed comincerete a leggere gli articoli di questo numero di rivista *Commodore* e tutti i numeri futuri, noterete che i listati dei programmi vengono riportati esattamente come li stampa la printer, e che tutte le simbologie speciali sono state modificate in maniera tale che il lettore interessato ad un programma non sia costretto ad «aguzzare» gli occhi per poter interpretare questo o quel simbolo.

Troverete allora al posto del cuoricino in negativo, che come sapete vuol dire pulisci lo schermo, questa espressione: [CLEAR]; oppure la lettera Q in negativo, cursore in giù, espressa con: [DOWN]; e così via per tutti i simboli di comando.

Potrete notare, laddove esistesse la necessità, che all'interno delle parentesi quadre può esistere anche un numero: infatti se in programma ci fossero sette Q negative, nel listato apparirà [7 DOWN].

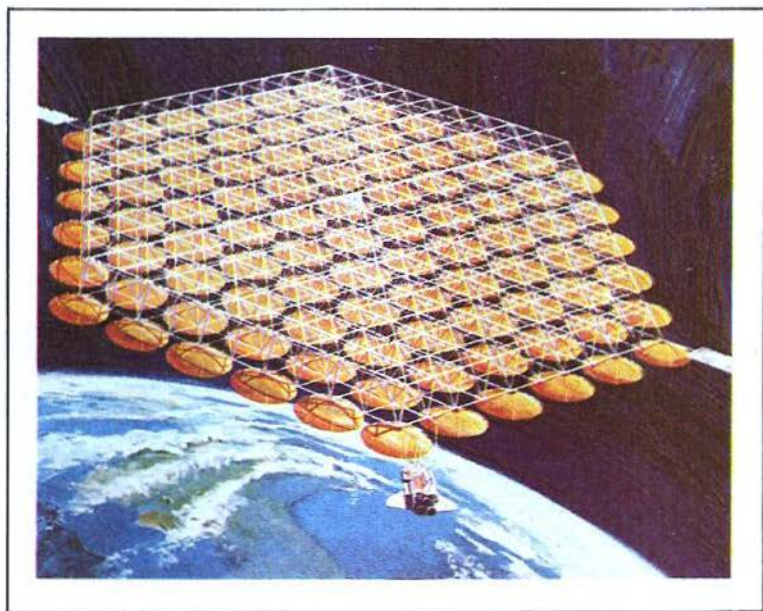
Qui di seguito viene riportata una tabellina di cui è inutile spiegare il significato. Tenetela sempre a portata di mano.

GRAFICO COMMODORE	VALORE ASCII	LISTATO	GRAFICO COMMODORE	VALORE ASCII	LISTATO
				5	[BIANCO]
				28	[ROSSO]
				159	[AZZUR]
	17	[DOWN]		156	[VIOLA]
	18	[RVS]		30	[VERDE]
	19	[HOME]		31	[BLEU]
	29	[RIGHT]		158	[GIALLO]
	20	[DEL]		129	[ARANC]
	145	[UP]		149	[MARR]
	146	[RVOFF]		150	[ROSA]
	147	[CLEAR]		151	[GRIGIO1]
	148	[INST]		152	[GRIGIO2]
	157	[LEFT]		153	[VERDE2]
π	255	[PI]		154	[CELESTE]
	144	[NERO]		155	[GRIGIO3]

Ah... dimenticavo... Grazie di aver comperato *Commodore!* Ne sarete sempre più soddisfatti.

Gloriano Rossi

FLUSSI RELATIVI

di Gloriano Rossi (i 2 KH)

Qualche anno fa, quando i prodotti Commodore non venivano commercializzati direttamente dalla Commodore Italia, ma da una azienda italiana, curavo per questa ditta una piccola rivista, un house organ chiamato "Pocket Pet".

Furono pubblicate quattro edizioni, di cui due doppie, con una tiratura di 5000 copie, cifra notevole per quei tempi.

Proprio su uno di questi numeri avevo trattato l'argomento dei flussi relativi.

Ancora oggi, e soprattutto oggi, ritengo valido riparlare di questi tipi di files di disco aggiornando quelle parole che avevo scritto allora.

Le unità Commodore

Una delle migliori caratteristiche dei prodotti Commodore risiede nel fatto che tutto è compatibile, o quasi. Infatti se io prendo un programma, scritto naturalmente in BASIC su un qualsiasi computer della CBM e lo carico su un altro qualsiasi computer, sempre Commodore, sono sicuro che "gira".

Quel "quasi" sta nel fatto che un Commdore 64 non possiede quei comandi specifici del BASIC 4. x, oppure esistono dei problemi per ciò che concerne alcune routine in linguaggio macchina e alcune locazioni di memoria.

A conferma di ciò possiamo notare che esistono innumerevoli programmi per il C64 che arrivano, senza alcuna modifica, direttamente dal Pet 2001, dal CBM 3032 dal 4032. Queste macchine, infatti, avevano caratteristiche estremamente similari al C64: il BASIC, le 40 colonne di video, l'indirizzamento alle periferiche, ect.

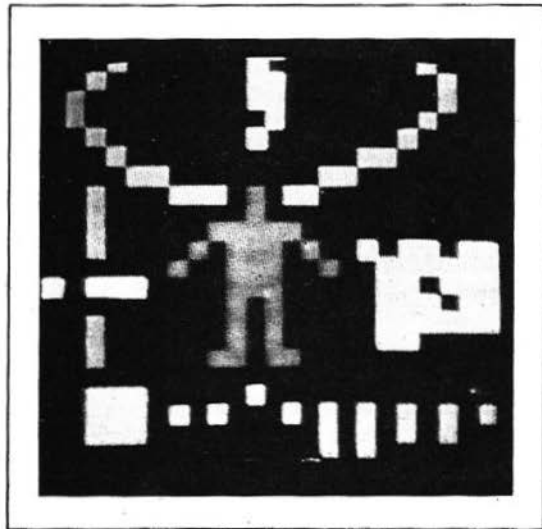
Ecco perchè il Commodore 64 è nato già con il piede giusto: Prima ancora di nascere aveva già un

numero elevato di programmi sia di giochi che gestionali. Solo dopo la sua nascita, le centinaia di programmi esistenti, sono diventate migliaia, proprio per le caratteristiche di potenza che portano il C64 ad un livello nettamente superiore rispetto ai predecessori.

Tutto quanto appena detto vale naturalmente anche per le periferiche. Infatti un dischetto formattato da una unità floppy disc CBM 1541, può essere tranquillamente letto da una unità CBM 2040, o CBM 2031, o CBM 3040, o CBM4040 e... naturalemente anche viceversa.

Ecco anche perchè quel mio scritto è ancora valido, ed ecco perchè ve lo ripropongo con le dovute modifiche ed aggiunte.

Prima di iniziare a parlare direttamente dei feles relative, vediamo quanti e quali tipi di flussi possono essere scritti e letti sulle unità disco Commodore 1540, 1541, 2031, 2040, 3040, 4040, 8050, 8250.

•**FLUSSO DI PROGRAMMA PRG** (tutte le versioni BASIC e di DOS) Contiene tutto il programma oggetto così come si trova nell'unità centrale (2001 - 3032 - 4032 - 8032 - VIC 20) la momento della SAVE; il file contiente anche il puntatore specifico del programma stesso, l'indirizzo cioè di dove dovrà essere riposto in memoria centrale al momento del LOAD.

•**FLUSSO SEQUENZIALE, SEQ** (tutte le versioni di BASIC e di DOS). Questo tipo di file è il più classico dei flussi trattati con qualsiasi elaboratore (anche non Commodore, naturalmente).
Un record viene scritto uno dopo l'altro e parimenti un record viene letto uno dopo e l'altro, proprio come con un nastro magnetico.
Tutti i records possono avere una propria lunghezza differenzte dalle lunghezze degli altri records del medesimo file e non esiste alcun identificatore si sistema relativo al singolo records.
Esiste esclusivamente un separatore (CHR$ (13)) per identificare la fine del record e l'eventuale inizio del record successivo.

•**FLUSSO RANDOM (USR)** o accesso diretto (tutte le versoni BASIC e di DOS). La caratteristica principale di un flusso organizzato con il sistema RANDOM è quella di poter accedere a qualsiasi record direttamente tramite l'uso di una chiave, senza per questo dover essere costretti a leggere sequenzialmente il file e quindi testare ogni record fino a trovare quello voluto. Un flusso organizzato con il sistema RANDOM è costituito in realtà da due files. Il primo, quello più grosso e capiente, contiene tutti i records scritti sequenzialmente, cioè uno dopo l'altro. Il secondo è molto più piccolo ed il suo contenuto è costituito dalle chiavi di ogni record del fiel principale, con in più le coordinate di posizione. Queste coordinate sono espresse in traccia, settore e posizione del byte di inizio record.

In sintesi viene riportato in questo file il numero del blocco ed il puntatore ove risiede o inizia il record interessato.

Per tutte queste ragioni un file organizzato in questa maniera è chiamato più propriamente I.S., da Indexed Sequential, poichè i record vengono scritti sequenzialmente, ma ciò nonostante si può accedere ad essi direttamente tramite indice.

A partire da questo tipo di organizzazione, il singolo record, o i campi relativi ad una informazione completa, devono avere una lunghezza ben precisa.

Se noi definissimo un flusso anagrafico organizzato in maniera I.S. dovremo dire innanzitutto quanto è lunga la chiave, poi quanto spazio occupa l'informazione e quindi quanto grande (numero records stimato) dovrà essere il flusso.

Il vincolo di lunghezza è dovuto principalmente dalla necessità di poter eseguire degli update (aggiornamenti) dei record, e quindi del file stesso. Infatti, se noi ad esempio non definissimo la lunghezza record (cosa comunque non permessa) potremo esclusivamente leggere o scrivere le inforamazioni, ma non potremo riscrivere un record dopo una correzione.

A seguito di un aggiornamento, un update, la lunghezza di un record potrebbe risultare più grande rispetto la versione precedente e quindi, momento della riscrittura, si verificherebbe un accavallamento con il record successivo con conseguente perdita o danno di quest'ultimo.

• **FLUSSO RELATIVE, REL** (DOS 2. x e BASIC 4 oppure BASIC C64). Dopo aver capito o quantomeno assimilato i concetti base del flusso SEQuenziale e del flusso I.S. possiamo passare alla descrizione del flusso RELative.

Forse è più interessante questo tipo di organizzazione di un file rispetto a quelle precedenti in quanto offre maggiori vantaggi rispetto alle organizzazioni SEQuenziale e indexed, soprattutto in determinati tipi di applicazioni. La differenza con un SEQuenziale stà nel fatto di poter accedere ai records sia sequenzialmente che in maniera diretta.

La differenza con un flusso RANDOM, sta nella mancanza del piccolo file delle chiavi e posizioni, nonchè per il tipo di scrittura che viene utilizzato; in realtà esiste sempre un piccolo file che fornisce al DOS (Disc Operating System, sistema operativo del disco) l'indirizzo fisico di ogni singolo record. Questo file è però nascosto, o per meglio dire è gestito direttamente dall'unità disco. Questo fatto ci permette di scaricare questo onere dall'operatore e quindi dalla programmazione.

Come si apre un flusso RELative su un sistema Commodore?

Sui manuali d'uso delle varie unità disco, è spiegato il sistema più semplice per l'opportuna apertura di un file RELative. Le spiegazioni, forse, non sono troppo chiare e quindi vediamo insieme:

OPEN 1,8,2, "nome del file, L, "+CHR$ (LR)

Il numero 1 sta per l'internal file number, numero interno del file; è proprio questo numero che serve al programma quando si esegue una PRINT # o una INPUT# quale diretto riferimento all'apertura. Quindi se scrivessimo 87 al posto dell'1 tutte le PRINT# o le INPUT# dovranno avere questo numero.

Il numero 8 indica qual è la periferica alla quale si riferisce la OPEN; in questo caso è naturalmente l'unita a dischi floppy.

Il numero 2 sta ad indicare l'indirizzo secondario (secondary address) che ci servirà, in particolare per i files relative, quale riferimento per il puntamento al record, vedremo comunque questo concetto in dettaglio più avanti.

In nome del file dovrà essere un insieme di lettere e/o cifre per un massimo di 16 caratteri.

Nei files sequenziali si aggiunge a questo punto generalmente ", S,R," per l'apertura sequenziale in lettura e ",S,W," per una OPEN di scrittura.

Nel caso dei files relative non si deve specificare se il file è aperto in Read o in Write, ma è sufficiente definire: ",L," con l'aggiunta della lunghezza del record, espressa però come CHR$, nell'esempio è stata posta una variabile LR proprio per indicare la lunghezza di ogni singolo Record, ad esempio 80.

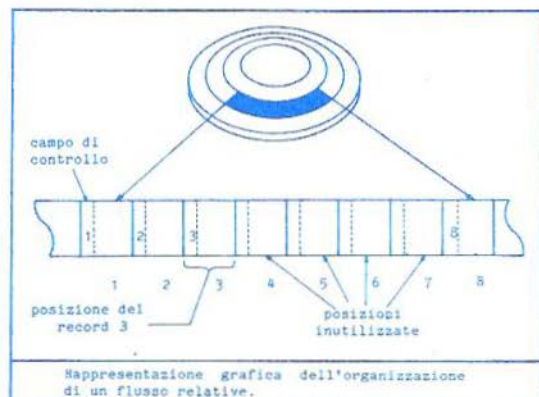
Con il BASIC 4. n tutto è ciò alquanto più semplice, infatti:

DOPEN# 1, "nome del file", L80

DOPEN # è un comando specifico del BASIC 4, infatti ci permette di aprire un file su disco senza dover specificare che si tratta di unità floppy e nè tantomeno di definire il Secondary Address.

Nome del file ha la stessa prerogativa che ab-

campo di controllo
1 2 3 4 5 6 7 8
posizione del record 3
posizioni inutilizzate

Rappresentazione grafica dell'organizzazione di un flusso relative.

biamo già visto.

La lettera "L" seguita da un numero definisce la lunghezza record in questione.

Ma torniamo al concetto che dice il flusso RELative è un file nel quale ogni record occupa una specifica posizione su disco, detta posizione direttamente legata per logica al record stesso. Con l'organizzazione RELative è consentito l'accesso diretto ad ogni singolo record, senza dover essere costretti ad esaminarne altri (SEQuenziale) oppure senza dover effettuare ricerche preventive in un indice (RANDOM o I.S.).

Numero relativo del RECORD

L'accesso e la scrittura di ogni singolo record posto in un file del tipo RELative sono effettuate tramite l'indicazione della posizione relativa del record nell'ambito del file. Per meglio spiegarmi dirò che la meccanica o meglio la logica di indirizzamento è uguale a quella che adottiamo in memoria centrale, nella RAM, quando abbiamo dimensionato una matrice di N elementi. Ogni informazione (variabile numerica o alfanumerica) occupa una posizione definita numericamente nella tabella.

Questa posizione è calcolata a partire dall'inizio del flusso stesso e a differenza delle DIM, non si ha la posizione zero, ma si parte sempre dalla posizione definita con il numero uno.

La posizione relativa di un record prende il nome di "NUMERO RELATIVO"; questo numero non è un indirizzo di disco, ma è un valore intero e positivo che viene tradotto dal sistema nell'indirizzo del record che è stato richiesto o che deve essere scritto o riscritto.

Il Numero Relativo di Record potrà assumere un valore che va da 1 a 65535, sempre che la capacità e le caratteristiche del file rispetti determinate regole. Infatti il DOS al momento della creazione di un file relative prepara anche un massimo di 6 blocchi di disco dedicati all'indirizzamento diretto ai blocchi di dati, ogni blocco di può puntare direttamente a 120 blocchi di dati; quindi i blocchi dei dati potranno essere al massimo 720, infatti 6 ·120=720, e sapendo che ogni blocco di dati può contenere un massimo di 254 caratteri (256 caratteri del blocco meno 2 di link= 254) potremo dire che un file relative potrebbe avere una capacità massima di 182.880 caratteri.

É chiaro che questo TOP, questo numero massimo, potrà essere raggiunto con unità disco tipo 8050 o superiori, mentre per le periferiche più usuali, tipo 1541 o 4040, è la capacità del dischetto che limita le dimensioni del file (167.432 caratteri).

Vediamo il perchè di questo numero:

La capacità massima di un dischetto è di 683 blocchi (683·256=174.848) di cui 19 sono dedicati alla directory, quindi ci rimangono 664 blocchi disponibili per i dati, ma ricordiamo che per ogni blocco è possibile memorizzare 254 caratteri, quindi 658·254=167.132.

A questo punto possiamo modificare, o meglio adeguare, l'affermazione di prima riguardante il numero massimo di caratteri per disco.

Allora per calcolare quanti record possono esistere in un file relative occorre sapere che alla lunghezza definita di ogni record si deve aggiungere 1 in quando il DOS separa un record da un altro con un CHR$ (13) (il RETURN).

Ora, se ho definito una lunghezza record di 9 caratteri dobbiamo calcolare 167.132 diviso 10 (9+1) per ottenere esattamente 16.713 record possibili.

É bene ricordare anche che la lunghezza massima di un record non potrà superare i 254 caratteri corrispondente alle dimensioni massime di un blocco di dati.

Il numero relativo di un record può essere parte del record stesso, oppure la risultante di determinate operazioni eseguite sul record o sui campi che lo compongono.

Un sistema estremamente facile per determinare il numero relativo di un record è quello di far coincidere questo numero con la chiave del record stesso. Se ad esempio, dobbiamo memorizzare un certo numero di articoli, il codice dell'articolo potrebbe corrispondere al numero relativo.
Così l'articolo il cui codice corrisponde al numero 005 occuperà la quinta posizione nel file in oggetto.

Ma come si fa a puntare direttamente al singolo record?

In BASIC 4 la dinamica è ancora una volta semplice:

```
RECORD# n, (nR)
INPUT# n, X$
```

L'istruzione RECORD#, seguito dal medesimo numero della DOPEN, ci permette di puntare direttamente al record richiesto; quindi l'istruzione IN-

PUT#, seguita dal medesimo numero della DOPEN, ci permette di leggere il record "puntato". Nella medesima maniera si scrive un record nella sua posizione relativa:

RECORD# n, (nR)
PRINT# n, X$

Il Commodore 64, può certo utilizzare il BASIC 4, ma solo come opzionale, mentre normalmente occorre sfruttare certi accorgimenti che il manuale d'uso ci suggerisce. Vediamoli:

In testa al programma, o comunque vicino alla OPEN del file relative, occorre aprire ancheil canale per i comandi diretti, in questa maniera:

OPEN 15,8,15

A questo punto l'istruzione RECORD si trasformerà:

NH = INT (nR/256)
NL = nR - NH · 256
PRINT# 15,"P" CHR$ (SA) CHR$ (NL) CHR$ (NH)
Con questa istruzione che cosa abbiamo detto al DOS?
Punta ("P") al record numero NH·256+NL del file che ho aperto con indirizzo secondario (SA, Secondary Address).

NH rappresenta la parte alta del numero record, mentre NL è la parte bassa dello stesso numero.

Ma perchè abbiamo trasformato il numero record in questa maniera? Per poter inviare l'informazione relativa al numero record all'unità disco, al DOS, possiamo utilizzare solamente la forma di CHR$ e quindi sapendo che un CHR$ può contenere solamente valori compresi fra lo 0 e il 255 dobbiamo sfruttare il sistema del Byte basso e del Byte alto, infatti provate con questo semplice programmino:

```
10 INPUT "NUMERO RECORD  ";NR
20 IF NR)256*256 THEN 10
30 NH=INT(NR/256)
40 NL=NR-NH*256
50 PRINT "IL VALORE ALTO E'  "NH
60 PRINT "IL VALORE BASSO E'  "NL
70 PRINT "INFATTI ";
80 PRINT NH;"* 256 +";NL;" = ";
90 PRINT NH*256+NL
```

Tutto è facile se una ditta possiede ad esempio mille articoli e li ha numerati progressivamente da 1 a 1000; il flusso da creare sarà costituito da 1000 porzioni di disco. Generalmente però il numero di codice di un articolo ben difficilmente parte dal numero 1 e prosegue in avanti. A maggior ragione raramente i numeri di codice degli articoli sono assegnati sequenzialmente e non tutti i numeri vengono assegnati a relativi articoli.

Avremo ad esempio una ditta che produce 800 articoli i cui codici vanno dal numero 00001 al numero 2400. Sarà illogico assegnare 2400 porzioni di disco per contenere quegli ottocento records; tre volte tanto il necessario. Per questo caso, ed altri analoghi, si utilizza una tecnica detta di randomizzazione.

La Tecnica di RANDOMIZZAZIONE

In pratica non si può dire che c'è una sola tecnica di randomizzazione. Ogni sistema atto ad individuare un numero relativo di record può essere chiamato tecnica di radomizzazione.

Vediamo una tecnica, alquanto semplice che si adatta in maniera egregia con l'esempio riportato: dividiamo il numero di codice per tre, infatti, guarda caso, 2400/800 dà proprio 3.

Il valore intero risultante dalla divisione corrisponderà al numero relativo di posizione del record nell'ambito del file.

Ogni tecnica di randomizzazione, però, genera o può generare inequivocabilmente dei records sinonimi. Ben difficilmente, infatti, potremo avere un numero veramente unico assegnato ad ogni record.

Vediamo un esempio

num. codice	costante	risultante	posizione
260	3	= 86.6666	86
·261	3	= 87	87
·262	3	= 87.3333	87
·263	3	= 87.6666	87
264	3	= 88	88

Che cosa è successo? I numeri di codice segnati con l'asterisco occuperebbero il medesimo posto nell'ambito del file; cosa naturalmente impossibile.

E se il campo della chiave non è un numero, ma una stringa di caratteri, cosa dobbiamo fare? Anche in questo caso occorre creare una routine di randomizzazione eseguendo alcuni algoritmi sulle lettere della stringa in esame.

La routine che riporto in esempio possiede l'unico difetto di aver bisogno di un file dimensionato il doppio del necessario, questo fatto però ci permette di avere pochissime possibilità di sinonimi.

```
  1 MF=4999: REM MAXFILE (ESPRESSO
    CON UN NUMERO PRIMO PIU' VICIN
    O AL NUMERO MASSIMO RICHIESTO
  2 FOR I=1 TO 16: SP$=SP$+CHR$(32
    ): NEXT: REM STRINGA DI SPAZI
 10 INPUT "LA STRINGA  ";X$
 11 X$=LEFT$(X$+SP$,10): REM PREND
    E IN CONSIDERAZIONE SOLAMENTE
    I PRIMI DIECI CARATTERI DELLA
    STRINGA.
 20 REM LA STRINGA X$ DEVE ESSERE
    DI LUNGHEZZA PARI
100 FO=0
105 FOR K=1 TO LEN(X$)
110 :FO=FO+((ASC(MID$(X$,K,1))-32)
    *1000)
120 NEXT K
130 K1=(FO/MF)*(FO/MF)
160 K1=ABS(K1)
170 IFK1)MF THEN K1=K1-MF: GOTO160
200 PRINT "IL NUMERO RELATIVO ="K1
```

Ma anche con questo tipo di randomizzazione possiamo avere, come già detto, dei sinonimi dovuti più facilmente alla quantità dei caratteri presi in considerazione piuttosto che alla validità dell'algoritmo stesso.

Che cosa fare allora in questo, nel precedente ed in altri casi analoghi? Tutte le tecniche che si possono utilizzare in questi frangenti si riassumono in unico termine:

La Gestione dei SINONIMI

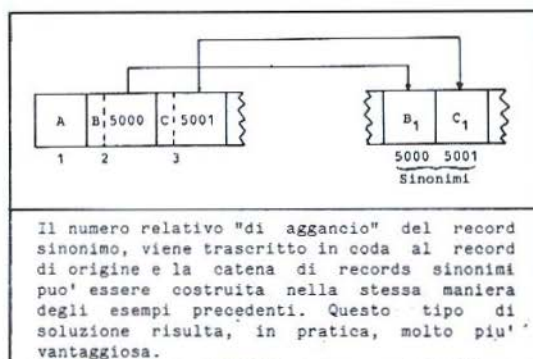
Il sistema più utilizzato in programmazione è quello di dimensionare il file RELative con uno spazio di circa il 15% maggiore rispetto allo spazio in realtà necessario (nel nostro caso 800 x 1.15 = 920, arrotondato per eccesso 999).

figura n. 2

Il migliore metodo per la gestione dei sinonimi e' quello di riservare una certa zona del flusso a loro dedicato

Se dobbiamo scrivere il record con il codice 262 e ci troviamo con la posizione 87 già occupata dall'articolo 261, dovremo riscrivere il record 87 con un campo in più che definirà la Nuova chiave di collegamento (maggiore di 800, nel nostro esempio).

figura n. 3

Il numero relativo "di aggancio" del record sinonimo, viene trascritto in coda al record di origine e la catena di records sinonimi puo' essere costruita nella stessa maniera degli esempi precedenti. Questo tipo di soluzione risulta, in pratica, molto piu' vantaggiosa.

In fase di lettura si dovrà inevitabilmente testare se il record, o meglio la chiave, posto nella posizione calcolata sia effettivamente quello voluto. In caso contrario si va ad individuare la chiave di collegamento che ci porterà in una nuova posizione del flusso.

Per certi tipi di applicazione, però, la zona dei sinonimi potrebbe risultare insufficente nonostante la valida capienza del file. Un caso tipico con questo problema è quello di una gestione di nomi e relativi numeri telefonici.

Il tipo di chiave che si vuole mantenere è quello relativo al nome.

Potrebbe verificarsi che dopo la traduzione del nome

il numero relativo si abbiano dei records sinonimi in quantità statisticamente superiore alle previsioni.

figura n. 4

Esempio grafico di flusso relative senza records sininimi.

Si scrivono allora i records nelle rispettive posizioni calcolate, fino a quando non si trovi il posto occupato.

figura n. 5

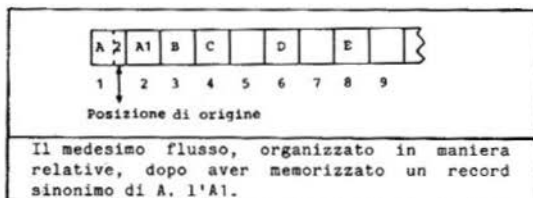
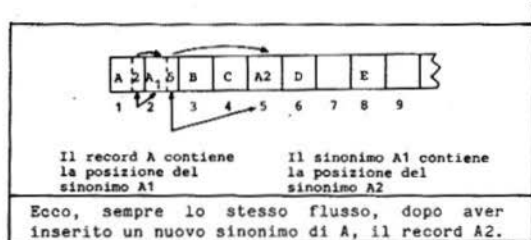
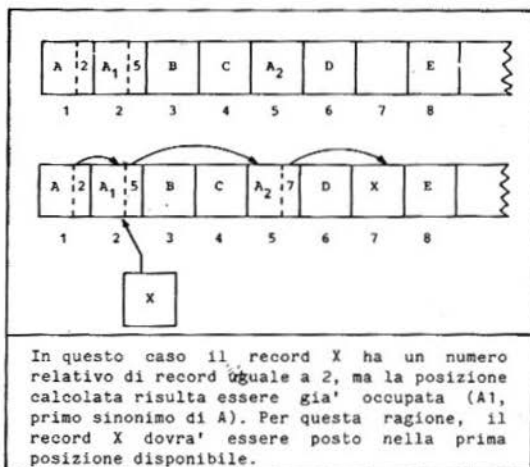
Il medesimo flusso, organizzato in maniera relative, dopo aver memorizzato un record sinonimo di A. l'A1.

Quando ciò avviene si ricerca la posizione libera più vicina e quindi si esegue la registrazione. Infine si va a scrivere il numero di aggancio in coda al record precedente, il capo-catena.

figura n. 6

Ecco, sempre lo stesso flusso, dopo aver inserito un nuovo sinonimo di A, il record A2.

Utilizzando questo sistema di gestione dei sinonimi, può accadere che un record debba occupare una posizione (primaria) occupata però da un elemento di una catena di sinonimi.

Per ovviare a questo inconveniente si va a ricercare l'ultimo elemento della catena "usurpante" e si registra il "malcapitato" nella posizione libera più vicina, riportando sempre il numero relativo nel record precedente. Avremo trattato il record "primario" allo stesso pari di un record sinonimo della catena.

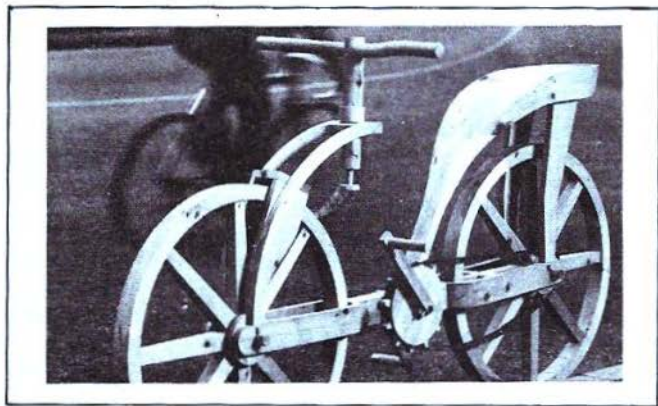
figura n. 7

In questo caso il record X ha un numero relativo di record uguale a 2, ma la posizione calcolata risulta essere gia' occupata (A1, primo sinonimo di A). Per questa ragione, il record X dovra' essere posto nella prima posizione disponibile.

Una breve conclusione

L'uso dei flussi relative visti in questa luce è da considerarsi terreno valido per i soli programmatori esperti, senza con questo sminuire o demoralizzare chi è alle prime armi, ma anzi deve essere un incentivo valido per migliorarsi e creare programmi sempre più sofisticati.

É sempre e soprattutto con l'esperienza e numerose prove precedute da analisi approfondite che è possibile eseguire e sfruttare validamente una gestione di files relative.

STARTER KIT

di **Giancarlo de Cobelli**

L'ORGAN, in ogni caso ci permette già di divertici discretamente. L'unica limitazione, è logico, risiede nella mancanza di una vera tastiera.

EFFETTI SONORI Questi mostrano degli esempi, sia pur banali, sull'uso del SID e dato che sono sempre scritti in BASIC possono essere riutilizzati per rendere i propri programmi più "gustosi".
Diversi premendo un tasto qualsiasi si cambia il colore di fondo.

Il programma di per se non è un gran che, ma può essere utile per sintonizzare perfettamente il monitor o televisore nonchè il trimmer di regolazione fine del colore posto al fianco dell'uscita video per televisore.

CHAR BOOT Permette la creazione di un set di caratteri personali ed utili ai propri scopi.

In confronto al programma presentato su questo stesso numero della rivista, il CHAR BOOT possiede più comandi; permette di spostarsi in qualsiasi posizione oppure di ruotare il carattere creato nonchè possibile memorizzare e richiamare i sets di caratteri creati per utilizzarli in altri programmi.

HIRES LOADER Mostra in alta risoluzione il volto di una famosissima attrice. Ecco un ulteriore esempio delle grandissime capacità del 64. Peccato che l'esempio non sia stampabile.

SPRITE BOOT Permette di creare uno o più sprite sullo schermo (piccoli disegni che si possono muovere sul video).

I disegni sono definibili punto per punto, su di una griglia di 21 linee per 40 colonne, automaticamente creata.

Una volta create, con gli appositi comandi, gli sprites possono essere memorizzati per utilizzarli con futuri programmi in BASIC od Assembler.

Ogni sprite usa 64 byte di memoria ed è usualmente chiamata pagina. Con questo programma possiamo definire 150 sprite poste fra le locazioni di memoria da 2047 a 4059 (pagine 32-63) e da 8132 a 16383 (pagine 128-255). Per poter sfruttare gli sprites, in BASIC, si deve utilizzare questa semplice routine:

```
10 REM LOAD SPRITES
20 IF SW =1 THEN 50
30 SW =1
40 LOAD "SPRITEFILE", 8,1
50 REM CONTINUAZIONE PROGRAMMA
```

La variabile SW funziona da switch o da flag e ci permette di sapere, se uguale ad uno, se sono già state caricate le sprite.
Gli sprite possono: essere creati nel modo alta risoluzione o nel modo multicolore. Possono essere espansi in orizzontale e/o in verticale. Essere mossi sullo schermo per vedere l'effetto visivo.

EDUCATIVI

BITS AND BYTES Semplicissimo

programma che propone i fondamentali concetti del computer. Per esempio: bit, byte, peek, poke, esadecimale e binario. Dopo alcune schermate di lezione pone delle domande per vedere se è stato capito il concetto. Fa uso di una buona schermata iniziale ed è utilissimo per chi voglia capire qualcosa su bit e byte.

DISK COMMANDS Già rivolto ad un campo specifico, questo programma illustra l'utilizzo del drive nei suoi comandi principali in ambiente DOS 5. 1 e Basic 4. 0. Fornisce un ampia ed esauriente spiegazione di tutti i comandi rivolti alla unità disco.

SNOOPY MATS Gioco educativo rivolto ai giovanissimi, permette loro di imparare la matematica giocando. Il tutto consiste nel fornire al computer la differenza numerica che esiste tra il barone rosso e snoopy. Diversi i livelli di difficoltà.

A conclusione di questa lunga chiacchierata possiamo dire che il pakage offerto dalla Commodore da utilizzare sul 64 è senza dubbio un ottimo prodotto in particolare rivolto agli operatori che sono sia alle prime armi, sia con qualche esperienza.

Ad eccezione di alcuni programmi veramente elementari, si può affermare che nel kit vengono offerti anche programmi che possono essere considerati di grande utilità.
Un nuovo package permette di toccare con mano e verificare le enormi capacità del Commodore 64. Si chiama Software Kit e contiene numerosi programmi applicativi e dimostrativi:
1) giochi di intrattenimento;
2) utility, ossia programmi che permettono di sfruttare alcune potenziali della macchina, alcuni dei quali possono considerarsi molto interessanti;
3) programmi dimostrativi di musica e grafica;
4) programmi educativi che aiutano a capire il funzionamento del computer.

Il package fornito su un supporto di dischetto è dotato di un ricco manuale di spiegazioni; è anche disponibile una versione su cassetta. In questo caso alcuni programmi, prettamente dedicati all'unità disco, non ci sono.

Inserendo il disco e digitando LOAD" 8*", e premendo il tasto RETURN carichiamo il primo programma presente sul disco, in questo caso il "C64. MENÚ", del quale riportiamo una versione "italianizzata".

Dopo aver digitato RUN il video mostrerà la directory (lista dei programmi contenuti sul disco) ed in seguito potremo fare quattro tipi di scelte:
1) vedere la pagina seguente di directory, qualora esista;
2) creare un nuovo file sequenziale della directory del disco nel caso in cui si sia inserito un nuovo programma oppure quando creare un nuovo dischetto con altri programmi;
3) uscire dal programma;
4) caricare ed eseguire il programma scelto.

Le prime tre opzioni richiedono solo la battitura di un tasto. Infatti è sufficente premere uno dei tasti funzioni suggeriti in basso allo schermo. La prima opzione è chiaramente intuitiva. Infatti, se la lista dei programmi esistenti richiede più pagine (videate) è possibile visionarle una dopo l'altra e ritornare alla prima videata premendo nuovamente il tasto funzione 1 (f1).

La seconda opzione ci permette di utilizzare il programma su tutti i dischi in nostro possesso.

Cosa dobbiamo fare?
La prima operazione di salvare il programma "C-64 MENU" come primo della directory del nuovo dischetto. Quindi si salvano i vari programmi che dovranno risiedere sul disco ed infine, eseguendo il medesimo "C64 MENÚ", si crea con F3 un nuovo file "DIRECTORY" (SEQ), ignorando la lista che ci propone inizialmente.

La quarta opzione del programma ci permette di digitare il numero corrispondente al nome del programma che si vuol caricare ed eseguire. Battete quindi il numero corrispondente e premete il tasto RETURN quale conferma oppure il tasto DEL in caso di errore.

Una caratteristica, se vogliamo piacevole, del "C64 MENÚ" è quella di visualizzare la directory in ordine alfabetico.

Tutti i programmi il cui nome inizia con uno spazio non vengono listati. Ciò permette di ignorare quei programmi che richedono un caricamento diverso da quello eseguito dal "C64 MENÚ". Spesso, infatti, troviamo programmi che necessitano di un LOAD diverso dal solito (LOAD" nome programma", 8,1) che, per "girare", devono essere richiamati da altri programmi detti "LOADER". Per esempio un programma che richiede il caricamento in una locazione di memoria diversa da quella di un normale programma in BASIC (locazione $0800, $2049), va caricato con i comandi LOAD" nome del programma", 8,1 come già detto.

Giochi

I giochi di intrattenimento inseriti in questo package non sono eccezionali, ma forniscono alcuni esempi dimostrativi delle capacità della macchina. Eccone alcuni.

ARROW In questo programma il gioco consiste nel guidare un serpente sullo schermo senza mangiarsi la coda. I punti si realizzano colpendo i rettangoli che appaiono a caso sul video.

LABIRINTO. Tramite la tastiera bisogna riuscire ad uscire dal labirinto che può assumere diversi formati a seconda delle difficoltà del gioco. Le dimensioni del labirindo possono variare da 2x3 a 15x10. I comandi per muoversi all'interno del labirinto sono:
Tasto F - Muove uno spazio avanti (se possibile).
Tasto L - Gira a sinistra (90 gradi).
Tasto R - Gira a destra (90 gradi).
Tasto H - "Help". Quando si preme

"H" viene mostrata la vista del labirinto dell'alto visualizzando la nostra posizione corrente.

LEMONADE. É un gioco d'affari, se così può chiamare. Lo scopo del gioco è quello di guadagnare più soldi possibili durante le dieci settimane di durata dell'estate.

Dopo aver dato il RUN, viene mostrato uno stand (come grafica si poteva fare di meglio) ed appaiono alcune domande del tipo: "come ti chiami?" lo "zucchero ?" il "succo di limone concentrato?" "quanti bicchieri vuoi acquistare?".

Con una base finanziara ed un listino prezzi stabiliti dal programma dobbiamo decidere cosa acquistare e il prezzo di vendita della limonata. Quest'ultimo deve anche tener conto della temperatura ambientale (visualizzata sul video costantemente) la quale varia di settimana in settimana.

Il tasto "H" ci permette di ottenere le istruzioni (in inglese come tutto il manuale ed i programmi).

Utility

Le utility che ora verranno descritte sono quasi tutte molto importanti per l'utilizzo del computer Commodore.

CHANGE DISK Permette di cambiare il numero del disk drive (normalmente 8) per poter fare la copia di un disco tra due unità.

COPY - ALL 64É la conseguenza logica del programma appena citato. Dopo aver cambiato il numero ad una unità, bisogna caricare il Copy-All 64. Dopo aver dato il numero del sorgente e della destinazione, ad esempio 8 per il sorgente e 9 per il destinatario, possiamo scegliere quali programmi copiare; se viene digitato esclusivamente l'"." verranno mostrati i nomi e copiati tutti i programmi; con alcune lettere più "." saranno copiati tutti i programmi che iniziano con le lettere digitate; se non si usa l'asterisco il programma richiesto verrà duplicato sul disco di destinazione.

A questo punto verranno mostrati tutti i programmi prescelti; per decidere quali copiare, digitare "Y" per si, o "N" per no. Alla fine della copia : Risposta: "Y" o "N" a seconda dei casi.

Il programma chiederà se si vuole effettuare una nuova copia

1541 Disk Backup. Permette di copiare fisicamente tutto il disco senza avere due unità. Seguendo le istruzioni mostrate nel riquardo "OPERATOR INTERVETION" è possibile fare la copia di tutto il disco scegliendo tra due opzioni:

1) digitando la lettera "B", backup selettivo, il programma determina quali aree del disco sono scritte effettuando la copia solo di quest'ultime. (Questo è il più veloce fra i due metodi);

2) digitando la lettera "D", backup completo, si effettua la copia esatta di ogni traccia e settore.

Attenzione: il programma Backup esegue sempre la formattazione di un disco di iniziare a copiare quello originale.

DUMP. Questo programma ci mostra sullo schermo un programma che risiede su disco in formato esadecimale. Ogni riga di schermo conterrà 10 byte.

Sulla sinistra di ogni riga troviamo una numerazione decimale che ci fornisce la quantità dei bytes presenti fino a quel punto.

Esempio:
```
0000: 08 50 00 00 01 00 00 00 00
0010: 00 07 0F 00 04 1A 00 00 00
```

LOAD ADDRESS. Interessante utility che mostra l'indirizzo di caricamento di un programma residente su disco. Normalmente i programmi sul Commodore 64 iniziano dalla locazione di memoria 2049. Questa routine serve per sapere se un programma deve essere caricato normalmente od in maniera speciale (cioè LOAD" nome programma" ,8 oppure se si deve caricare con LOAD" nome del programma", 8, 1).

SUPERMON64. V1 Questo utility, molto interessante, permette di programmare il Commodore in linguaggio macchina. Con l'aiuto di uno dei numerosi libri che trattano del sistema operativo e del linguaggio assembler del microprocessore 6502 e 6510 potremo entrare nel fantastico mondo del linguaggio macchina che permette di sfruttare a pieno le possibilità del Commodore 64 (vedi i numerosi giochi in commercio che normalmente sono in LM).

Da notare che tutti i comandi devono essere seguiti dal tasto di RETURN e tutti i valori numerici devono essere inseriti in codice esadecimale.

PET EMULATOR. Molto utile questa emulazione del PET che permette di utilizzare sul Commodore 64 quasi tutti i programmi PET esistenti.

Normalmente il C64 carica programmi BASIC dalla locazione di memoria 2048 alla 40359 e gestisce lo schermo dalla 1024 alla 2048. L'emulatore riconfigura la memoria del 64 trasferendo l'area di memoria del BASIC alla locazione di memoria 1024 fino alla 32767 e la gestione dello schermo alla 32768 fino alla 36863.

Il programma va caricato con l'istruzione LOAD" PET EMUALTOR", 8,1 e fatto girare eseguendo il comando SYS 12'4096. La spiegazione del perchè dell'"1", nella sistassi di LOAD è già stato spiegato precedentemente, infatti il PET EMULATOR viene ed il numero della SYS corrisponde all'indirizzo di memoria da cui parte il programma. Ora tutte le POKE e le PEEK dello schermo, dell'area BASIC sono accesibili direttamente da tastiera. Le più usate di queste POKE sono:

Poke 59467, 16 = abilita il modo musica

Poke 59464, X = decide la frequenza

Poke 59466, Y = decide il range di frequenza
Poke 59468, Z = con Z=12 modo grafico
con Z=14 modo minuscolo

Inoltre tutte le POKE comprese tra gli indirizzi 0000 e 1023 quando possibile sono tutte utilizzabili. Le POKE non interpretate daranno un messaggio di errore del tipo 'illegal quantity error'. C'è da fare attenzione a caricare i programmi in linguaggio macchina perchè molti non li prende.

DOS WEDGE. Questo programma è anche chiamato DOS 5.1. ed è utilizzato per abbreviare di molto le comunicazioni con il drive. Esso provvede a tradurre i comandi BASIC in comandi abbreviati con una simbologia speciale che più avanti vedremo. Il programma si carica con le istruzioni LOAD" DOS WEDGE", 8,1 ed al posto del RUN la SYS52224.

Il DOS supporta i comandi del tipo (copy, scratch, rename, new a disk), un comando per leggere la directory ed i comandi per caricare e salvare i programmi. Il DOS può anche creare e mantere volumi di programmi (gruppi di programmi). Ogni comando iniza con un carattere seguito poi dal nome del programma così come mostrato:
il segno " " o " " sostituisce i seguenti comandi BASIC:

```
10 OPEN 15, 8,15
20 INPUT # 15, B$, C, D
30 PRINT # 15, B$, C, D,
```

@ $ (drive): (nome del programma) (*) ([volume])
Questo comando legge la directory del disco e la stampa sullo schermo.

Se viene specificato il nome del programma solo quel programma verrà mostrato; se alcune lettere seguite da "-" tutti i programmi che iniziano con quelle lettere, se solo "-" tutta la directory. Se il [volume] è specificato saranno mostrati solo i programmi contenuti nel volume.
@ N (drive): nome del disco, id
Formatta il nuovo disco con il nome e l'identificatore specificato.
@ R (drive): nuovo programma = vecchio programma
Sostituisce il vecchio nome del programma con in nuovo indicato.
S (drive): nome programma (-) [volume].
Cancella il programma specificato dal comando. Per "-" vedi sopra la descrizione.
@ UI (drive)
Questo comando resetta il DOS.
@ I (drive)
Inizializza il drive
@ Q
Con questo comando serve per disabilitare il programma DOS.
/ nome programma
Corrisponde in BASIC al comando LOAD" nome programma", 8
% nome programma
Corrisponde in BASIC ai comandi LOAD" nome programma", 8 1
FRECCIA IN ALTO nome programma
Corrisponde in BASIC al comando LOAD" nome programma", 8
FRECCIA A SINISTRA nome del programma
Corrispondente in BASIC al SAVE" nome programma", 8
É evidente che il programma DOS una volta caricato resta sempre in memoria anche se vengono caricati altri programmi, a meno che questi ultimi non partano dalla locazione di memoria 52224. Per cancellare in maniera definitiva il DOS WEDGE si utilizza la SYS64738, cioè il system reset da software.

AMORT TABLE. Questo programma fornisce una tavola di ammortamento dei prestiti di qualsiasi genere. Il programma pone cinque domande del tipo:
1) interesse rate annuo;
2) numero dei periodi di interesse per anno;
3) numero di pagamenti per anno;
4) ammontare del prestito;
5) durata del prestito negli anni;

Dopo aver risposto alle cinque domande vengono mostrate nelle colonne: il numero dei pagamenti, gli interessi sul debito, le quote di pagamento ed il debito insoluto.

L'unico neo di questo programma risiede nel fatto che è stato studiato per il mercato monetario canadese. Ma ciò non dovrebbe spaventare: infatti è stato scritto interamente in BASIC e quindi sono possibili le opportune modifiche.

MORTAGAGE. Questo programma calcola le informazioni essenziali al pagamento ed alla scadenza delle rate di una ipoteca.

Dopo aver risposto a tutte le domande poste dalla macchina verrà mostrato sul video l'ammontare degli interessi e pagamenti, l'ammontare mensile. Purtroppo anche per questo programma vale il discorso fatto prima per il "Amort Table".

CALENDAR. Calendario perpetuo dal 1752. Basta inserire data, mese, anno e.... lasciarlo sempre acceso.

MUSICA E GRAFICA

ORGAN Il programma dimostra le "mostruose" capacità del circuito integrato chiamato SID contenuto nel Commodore 64.

Dopo averlo caricato in memoria ed avere comandato il RUN, lo schermo mostra una tastiera con specificate le lettere da utilizzare, i comandi per cambiare le ottave (4) ed il suono (3 suoni più la batteria).

Con la sbarra di spazio scegliete il suono polifonico o d'assolo.
É vero! Anche in commercio sono reperibili dei sintetizzatori forse o sicuramente migliori, ma qui abbiamo un computer che fa egregiamente anche questo senza avere nulla a che invidiare ed apparati dedicati di costo uguale o anche leggermente superiore.

```
0  [reversed header]
8    " C64.MENU"            PRG
2    "   DIRECTORY   "      SEQ
4    "CHANGE DISK"          PRG
10   "COPY-ALL64"           PRG
2    "DUMP"                 PRG
2    "LOAD ADDR"            PRG
10   "SUPERMON64.V1"        PRG
3    "COLOR TEST"           PRG
4    "CHAR BOOT"            PRG
32   " CHAR EDITOR"         PRG
1    " ROTATE.DATA"         PRG
9    " STANDARD.SET"        PRG
9    " COMPUTER.SET 5"      PRG
4    "HIRES LOADER"         PRG
4    " VMSUPP"              PRG
32   " GARBO"               PRG
3    "SPRITE BOOT"          PRG
43   " SPRITE EDITOR"       PRG
2    " SCROLL.DATA"         PRG
3    " SAMPLE SPRITES"      PRG
9    "ORGAN"                PRG
7    "SONG - DIXIE"         PRG
2    "SONG - TWINKLE"       PRG
6    "SONG - YANKEE"        PRG
2    "SOUND - ALIEN"        PRG
2    "SOUND - BOMB"         PRG
2    "SOUND - GUNFIRE"      PRG
2    "SOUND - PONG"         PRG
2    "SOUND - RAYGUN"       PRG
1    "SOUND - SIREN"        PRG
1    "SOUND - CLAP"         PRG
32   "ARROW"                PRG
2    "DEMO JOYSTICK"        PRG
37   "LABYRINTH"            PRG
51   "LEMONADE"             PRG
47   "BITS AND BYTES"       PRG
5    " BYTSPRITES"          PRG
51   "SNOOPY MATH"          PRG
39   "AMORT TABLE"          PRG
45   "MORTGAGE"             PRG
38   "CALENDAR"             PRG
17   " PET EMULATOR"        PRG
4    " DOS WEDGE"           PRG
44   "DISK CMDS"            PRG
28   "1541 BACKUP"          PRG
0 BLOCKS FREE.
```

Figura n. 1
Contenuto della directory del disco fornito con il pacchetto STARTER KIT. É da notare che esiste un solo file sequenziale che contiene il direttorio completo dei programmi che non iniziano con lo spazio.

```
100 REM ********************************
110 REM *
    *
120 REM * AUTOMATIC MENU PER DISCO    *
130 REM *
    *
140 REM * (DEVE ESSERE SEMPRE IL PRIMO *
150 REM *  PROGRAMMA DI OGNI DISCO)   *
160 REM * SI CARICA CON    LOAD":*",8  *
170 REM *
    *
180 REM ********************************
190 POKE 53280,6:POKE 53281,6:DIM TB$(150):PRINT "[CLEAR]" CHR$(14):DI=8:C=1
200 OPEN 8,DI,8,":   DIRECTORY   ":OPEN 15,DI,15:INPUT#15,EN$,ED$,ET$,ES$
210 INPUT#8,DN$:IF EN$<>"00" THEN CLOSE 8:CLOSE 15:GOTO 540
220 INPUT#8,TB$(C):IF ST=66 THEN 250
230 IF  ASC( LEFT$(TB$(C),1))=10 THEN TB$(C)= RIGHT$(TB$(C), LEN(TB$(C))-1):GOTO 230
240 C=C+1:GOTO 220
250 CLOSE 8:CLOSE 15:M=C-1:TB$(C)="":L1$="[RVS]":FOR C=1 TO 40:L1$=L1$+" ":NEXT
260 L1$=L1$+"[RVOFF]":A=1
270 POKE 53281,1:PRINT "[CLEAR][BLEU]";:FOR C=1 TO 4:PRINT L1$;:NEXT :FOR C=1 TO 2:PRINT "[BLEU]"L1$;
280 NEXT :PRINT "[HOME]";
290 PRINT  SPC(3)"[RVS][BLEU] ........................... "
300 PRINT  TAB(5)".EVISIONE : APRILE 1984"
310 PRINT  SPC(5)"DISCO : [ROSSO]"DN$ SPC(3)L1$;
320 FOR C=A TO A+14:PRINT "[RVS]" RIGHT$("    "+ STR$(C),5)"[RVOFF] [BLEU]"TB$(C) SPC(29 - LEN(TB$(C)));
```

```
330 PRINT "[ROSSO][RVS]        ";:N
    EXT :PRINT L1$;
340 FOR A=1 TO 2:PRINT "[BLEU]"L
    1$;:NEXT :PRINT  LEFT$(L1$,4
    0)"[UP]"
350 PRINT  SPC(5)"[RVS][2 UP].1
    = PER CONTINUARE IL LISTATO"
    ;
360 PRINT  SPC(10)".3 = PER CREA
    RE LA NUOVA DIRECTORY";
370 PRINT  SPC(6)".5 = PER USCIR
    E DAL PROGRAMMA[BLEU][HOME]"
380 POKE 2023,160:POKE 562,56:PO
    KE 198,0:B$=""
390 GET A$:IF A$="" THEN 390
400 IF  ASC(A$)=13 THEN 490
410 IF A$="." THEN 470:REM " F1
    ******
420 IF A$="." THEN 530:REM " F3
    ******
430 IF A$="." THEN POKE 53280,6:
    PRINT "[CLEAR][3 DOWN][BLEU]
    SYS64738[HOME]":END :REM " F
    5 ******
440 IF  ASC(A$)=20 THEN B$="":PR
    INT "[HOME][4 DOWN][ROSSO][R
    VS]    [RVOFF]":B$="":GOTO 39
    0
450 IF A$<"0" OR A$>"9" OR  LEN(
    B$)=3 THEN 390
460 B$=B$+A$:PRINT "[HOME][4 DOW
    N][BLEU]"B$:GOTO 390
470 IF C>M THEN A=1:GOTO 270
480 A=C:GOTO 270
490 IF  VAL(B$)>M THEN PRINT "[H
    OME][4 DOWN][BLEU]???":FOR C
    =1 TO 1500:NEXT :A$= CHR$(20
    ):GOTO 440
500 POKE 53281,6:POKE 53280,14:P
    RINT "[RVOFF][CLEAR][3 DOWN]
    [BIANCO]LOAD" CHR$(34)TB$( V
    AL(B$)) CHR$(34)",8"
510 REM  SIMULAZIONE ESECUZIONE
    TASTO RETURN E ISTRUZIONE RU
    N E RETURN
520 PRINT "[HOME]":POKE 198,4:PO
    KE 631,13:POKE 632,82:POKE 6
    33,213:POKE 634,13:END
530 PRINT "[HOME][4 DOWN][BLEU]W
    AIT"
540 OPEN 8,DI,0,"$0":FOR C=1 TO
    8:GET #8,A$:NEXT :C=1:DN$=""
    :FOR C=1 TO 16
550 GET #8,A$:DN$=DN$+A$:NEXT :G
    ET #8,A$:GET #8,A$:DN$=DN$+"
     ":GET #8,A$
560 DN$=DN$+A$:GET #8,A$:DN$=DN$
    +A$:GET #8,A$:GET #8,A$:DN$=
    DN$+"  "+A$
570 GET #8,A$:DN$=DN$+A$:GET #8,
    A$:C=1
580 FOR A=1 TO 4:GET #8,A$:NEXT
    :PN$="":TY$=""
590 GET #8,A$:IF ST<>0 THEN 680
600 IF A$="" THEN 680
610 IF  ASC(A$)<>34 THEN 590
620 GET #8,A$:IF  ASC(A$)<>34 TH
    EN PN$=PN$+A$:GOTO 620
630 GET #8,A$:IF  ASC(A$)=32 THE
    N 630
640 TY$=TY$+A$:GET #8,A$:IF A$<>
    "" THEN 640
650 IF  LEFT$(TY$,3)<>"PRG" THEN
     580
660 IF  LEFT$(PN$,1)=" " THEN 58
    0
670 TB$(C)=PN$:C=C+1:IF ST=0 THE
    N 580
680 CLOSE 8:OPEN 15,DI,15,"S:
    DIRECTORY  ":CLOSE 15
690 Z$="................":OPEN 8
    ,DI,8,":  DIRECTORY  ,S,W"
    :PRINT#8,DN$
700 FOR A=1 TO C-1:C$=Z$:FOR B=1
     TO C-1:IF C$<TB$(B) THEN 72
    0
710 C$=TB$(B):D=B
720 NEXT :PRINT#8,C$:TB$(D)=Z$:N
    EXT :CLOSE 8:CLR :GOTO 190
```

Figura n. 2
L'AUTOMATIC MENU qui listato vi potrà servire per accedere in maniera semplice e organizzata ai programmi residenti sui vostri dischetti.

```
10 PRINT "[CLEAR]":POKE 53280,0:P
   OKE 53281,0
12 PRINT "[RIGHT][RVS][VERDE]INDI
   RIZZO DI PARTENZA DI UN PROGRA
   MMA"
15 PRINT  "[DOWN]PROGRAMMA [GIALL
   O]";
16 INPUT  F$
```

```
18 PRINT "[DOWN][VERDE]INDIRIZZO
   DI PARTENZA: [ROSSO]";
20 OPEN 1,8,3,"0:"+F$
30 GET  #1,L$
40 GET  #1,H$
50 X$=H$:GOSUB  100:H=X*256
60 X$=L$:GOSUB  100:L=X
70 PRINT  H+L"[VERDE]"
80 CLOSE 1
90 END
100 IF  X$=""  THEN  X=0:RETURN
110 X= ASC(X$)
120 RETURN
```

Figura n. 3
Questo breve listato di programma vi può fornire sempre l'indirizzo di partenza, in memoria, di un qualsiasi altro programma residente su disco.

```
  1 REM *************************
    *******
  2 REM *
          *
  3 REM * STAMPA ESADECIMALE DI FI
    LES    *
  4 REM *
          *
  5 REM *************************
    *******
 10 PRINT  "[CLEAR]"
 11 POKE  53280,0: POKE  53281,0
 12 PRINT   TAB(5)"[RVS][ROSSO]DUM
    P DI UN FILE IN ESADECIMALE[DO
    WN]"
 14 C=0: H$="0123456789ABCDEF": T=
    100000
 16 PRINT  "FILE";: INPUT F$: OPEN
     1,8,3,F$
 26 PRINT  "[ROSSO]"  RIGHT$( STR$
    (T+C),5);"[VERDE]: ";
 28 FOR  I=1  TO  10: GET #1,X$
 30 C=C+1: IF X$=""  THEN X$= CHR$
    (0)
 32 IF  ST  AND  64  THEN  90
 35 X= ASC(X$): H= INT(X/16): L=X-
    (H*16)
 38 PRINT   MID$(H$,H+1,1)+ MID$(H
    $,L+1,1);" ";
 41 IF  C/240= INT(C/240)  THEN  G
    OSUB  100
 42 NEXT : PRINT : GOTO  26
 90 CLOSE  1
100 GET  A$: IF  A$=""  THEN  100
110 RETURN
```

Figura n. 4
Questa utility visualizza in esadecimale un qualsiasi file residente su disco. Sarà sufficiente fornire il nome e sullo schermo apparirà il contenuto richiesto. É stata aggiunta una piccola routine di fermo quando il video è riempito.

```
1000 REM  **D*****************C**
1003 REM  ** JOYSTICK DEMO PRG **
1005 REM  **D*****************C**
1006 :
1007 PRINT  "[CLEAR]": POKE  5328
     0,0:POKE  53281,0
1010 GOSUB  2000: PRINT  TAB(12)
1020 IF  JV=1   THEN  PRINT "[VER
     DE]    A L T O   [DOWN]"
1030 IF  JV=2   THEN  PRINT "[ROS
     SO]   B A S S O  [DOWN]"
1040 IF  JV=4   THEN  PRINT "[GIA
     LLO]S I N I S T R A[DOWN]"
1050 IF  JV=5   THEN  PRINT "[ARA
     NC]ALTO & SINISTRA[DOWN]"
1060 IF  JV=6   THEN  PRINT "[CEL
     ESTE]BASSO & SINISTRA[DOWN]"
1070 IF  JV=8   THEN  PRINT "[BIA
     NCO]  D E S T R A [DOWN]"
1080 IF  JV=9   THEN  PRINT "[GRI
     GIO1] ALTO & DESTRA[DOWN]"
1090 IF  JV=10  THEN  PRINT "[MAR
     R]BASSO & DESTRA[DOWN]"
1630 IF  FR= 16  THEN  1010
1640 PRINT   TAB(12)"[VERDE]** [R
     OSSO]S [AZZUR]P [VIOLA]A [BL
     EU]R [GIALLO]O [VERDE]**[DOW
     N]"
1660 GOTO  1010
1990 :
2000 REM     *** JOYSTICK ROUTINE
      ***
2010 :
2020 POKE  56322,0
2030 JV= PEEK(56320)
2040 FR=JV  AND  16
2050 JV=15-(JV  AND  15)
2060 POKE  56322,255
2070 RETURN
```

Figura n. 5
Volete provare il vostro Joystick? Ecco il programmino ideale dal quale ricopiare, per vostri programmi, la routine di riconoscimento movimento (righe 2000 in poi).

UN LIST PIU' FACILE

di Tullio Spezia

(I 2 TZS)

Quante volte esaminando un programma direttamente sul video abbiamo espresso il desiderio: «Che bello se si potesse fermare il LIST e poi ripartire!».

Quando si studia un programma per correggerlo, si è forzati a fermare ogni poche linee lo scorrere del listato sullo schermo, semplicemente premendo lo «stop», ed ogni volta, di conseguenza, bisogna poi ripetere la richiesta al computer, digitando «LIST» seguito dal numero della linea da cui si vuole ripartire, ed eventualmente la lineetta per l'indicazione di continuare automaticamente con le linee successive.

Non sarebbe più bello fare tutto il LISTato con una unica richiesta trattenendo fermo il passaggio del programma con un tasto apposito (mentre con il CTRL lo si rallenta)?

Riferendoci al Commodore C-64, il pensiero va subito al tasto SHIFT/LOCK, quello che normalmente viene ignorato od usato al più occasionalmente nei word processor. Una volta azionato, se ne sta tranquillo ad aspettare di essere disinserito e ci lascia le mani completamente libere; inoltre è adiacente al tasto CTRL, che potremmo toccare con dito medio della sinistra, e con il dito indice inserire lo SHIFT/LOCK.

Per fare ciò occorre introdurre nel Commodore questa nuova funzione, non in BASIC, purtroppo, ma in linguaggio macchina.

Nonostante ciò, il programma che segue è stato scritto in BASIC per facilitarne l'utilizzazione, ed ha la particolarità di «nascondersi» una volta che viene dato il RUN (il NEW nell'ultima linea ha effetto solamente sulla parte in BASIC, ma non su quella in linguaggio macchina che a quel punto è già locata nella sua giusta posizione).

Sembrava una cosa tanto lunga ma invece questo è tutto.

Sicuramente questa piccola routine vi diventerà subito amica. Infatti è facilmente caricabile dal registratore o dal disco per la sua brevità, ed una volta che è nel computer, vi rimane finchè questo non venga spento o non subisca la SYS64738, il system reset.

Ecco il testo del programma:

```
10 A=232: B=PEEK(56)-1: C=PEEK(77
   5): IF C<167 THEN 110
20 POKE 55,A: POKE 51,A: POKE 56,
   B: POKE 52,B: POKE 774,A: POKE
   775,B
30 F=A+B*256
40 G=F+22
50 IF F=G GOTO 80
60 READ D: POKE F,D
70 F=F+1: GOTO 50
80 POKE F,C
90 DATA 72,152,72,32,159,255,169,
   1,44,141,2,208,246
100 DATA 169,0,133,198,104,168,104
   ,76,26
110 NEW
```

Ora il LIST è più facile vero? Allora... buon lavoro!

RTTY MANAGEMENT SYSTEM

di **Ambrogio Mainini (IW 1BBG) e Paolo Denaro (i 2YNA)**

Non poteva certo mancare, in questo primo numero di *Commodore* un programma interamente dedicato a quella strana categoria di radioamatori che, affascinati anche da quelli che possiamo definire "pruriti informatici", mani al portafoglio, hanno acquistato un home-computer: il 64.

Contrariamente a quanto possa sembrare a prima vista, ciò che proponiamo qui di seguito è un programma strutturalmente nuovo anche se i comandi operativi sono volutamente rimasti pressochè uguali (per comodità di apprendimento) a quelli usati da altri collaudatissimi programmi "sui generis".

Riteniamo opportuno sottolineare alcune particolari caratteristiche che rendono il programma RTTY più rispetto versatile ai suoi predecessori.

Caratteristiche

- Il programma prevede la memorizzazione di un testo, durante la fase di ricezione. La capienza di questo magazzino è di oltre 2500 caratteri.

Riteniamo che detta capacità è in grado di soddisfare il più logorroico degli "aficionados" della RTTY.

- L'editing del buffer di tastiera viene visualizzato sulle tre righe nella parte inferiore dello schermo, con scrolling orizzontale.

Questo sistema permette di evitare che uno scrolling normale metta fuori campo una riga intera di messaggio. Il messaggio così digitato potrà anche essere corretto prima di essere inviato in trasmissione.

- La chiamata ripetuta automaticamente (CQ CQ CQ DE.....) viene istantaneamente interrotta con la commutazione in ricezione di tutto il sistema allorquando vengono ricevute almeno quattro serie di RY.
- Per quel che riguarda l'uso dei mo-

Tabellina comandi in ricezione

F1
Commuta in trasmissione.
F2
CQ Automatico (Si blocca alla ricezione di almeno quattro coppie di RY oppure di un HOME (comando manuale).
F3
Menù velocità disponibili.
F4
Libero.
F5
Commuta il LTRS (in caso di QRM).
F6
Cambio BAUDOT-ASCII e viceversa.
F7 + N
(Numeri da 0 a 9). Memorizzazione di messaggi volatili o estemporanei (max 255 cad.).
Per volatile si intende una memorizzazione che si mantiene solamente fintantochè la macchina, il C64, rimane accesa, oppure non vengano sovrapposti altri messaggi.
F8
Libero.
Freccia a sinistra
Memorizzazione volatile del nominativo del corrispondente.
Lira sterlina
Cambia il testo al CQ automatico (detto cambiamento è volatile. Questa opzione è utile durante i Contest).

di di emissione e delle velocità ricordiamo che esiste una normativa ben precisa che ne regolamenta l'utilizzazione; pur tuttavia il programma consente la trasmissione e la ricezione in BAUDOT e in ASCII anche a velocità diverse da quelle standard.

Al fien di facilitare la scelta dei modi e delle velocià, è stato predisposto un menù speciale di rapidissima consultazione.

• Durante la trasmissione è possibile scegliere il modo di invio del messaggio:

a) Letter by letter. Ogni singola lettera viene trasmessa nello stesso istante in cui viene digitata.

b) Word model. La trasmissione è attivata dallo "Space" o dal "LINE FEED" per cui se la parola editata non è seguita da uno dei comandi citati, stazionando nel buffer di tastiera, può subire qualsivoglia modifica o correzione prima dell'emmissione.

Il sistema è stato dotato anche della capacità di trasmettere ciò che appare sul video. Questo fatto può diventare utile quando si voglia avere la possibilità di far "vedere" all'interlocutore come lo si riceve.

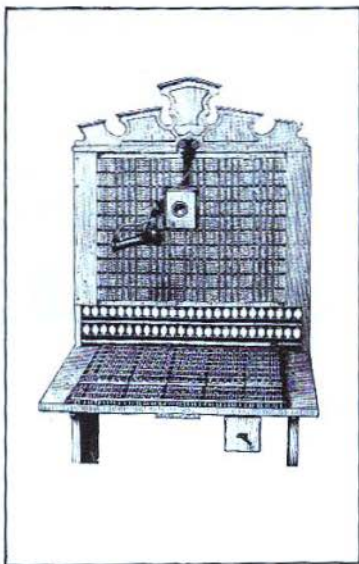
É stato previsto un controllo speciale per permettere sia la trasmissione che la ricezione di quel carattere "difficile" che corrisponde alle virgolette.

Infatti il medesimo problema si riscontra in vari Wordprocessors che prevedono, per questi casi, una gestione particolareggiata. Tutto ciò è dovuto al fatto che Commodore interpreta l'apertura delle virgolette quale prossimo contenuto di stringa accettando quindi alcuni comandi in maniera diversa da quella che si vorrebbe.

Al fine di rendere possibili eventuali aggiunte al programma sono stati lasciati liberi sia in trasmissione che in ricezione, alcuni tasti funzione.

Prima di vedere come è possibile collegare il Commodore al modem o demodulatore RTTY è doveroso e corretto fare due precisazioni:

1) la stesura del programma è opera di Ambrogio, con il quale Paolo ha tentato di collaborare;

2) il demodulatore (modem) RTTY

che è stato utilizzato da entrambi gli autori è l'ottimo TU 170-V progettato e distribuito da i2ZGP di Varese.

I collegamenti C64/Modem

I collegamenti da effettuare fra il Commodore 64 ed il demodulatore RTTY sono alquanto semplici, infatti sono necessari in tutto quattro cavetti che andranno saldati ad un connettore per la USER-PORT (connettore 12+12 a passo 3.96).

Sapendo che le connessioni della USER PORT sono disposte su due file (osservate il manuale del Commodore 64), si prendono in considerazione solamente le connessioni della parte inferiore quelle cioè contrassegnate dalle lettere che vanno alla A alla N.

I Pin A della USER PORT, corrispondente alla massa (sia logica che fisica) del C64, va collegato alla mas-

Tabellina comandi in trasmissione

F1
Commuta in ricezione inviando automaticamente il messaggio: PSE KKK.

F2
Libero

F3
Modo di trasmissione (carattere per carattere o parola per parola).

F4
Ora e data.

F5 + N
Dove N è un numero da 0 a 9. Emissione messaggi prememorizzati (vedi tabellina messaggi di F5)

F6
Libero.

F7 + N
(N è un numero da 0 a 9). Emissione messaggi estemporanei memorizzati con il comando F7 di ricezione.

F8
Emissione Videata (Si interrompe HOME) per ritrasmettere quando si è ricevuto.

Freccia verso l'alto
Emissione del nominativo del corrispondente con l'aggiunta di "de" ed il proprio nominativo.

Lira sterlina
Esegue l'emissione del numero 73 in grande.

HOME
Commuta istantaneamente in RX (ricezione).

DEL
Blocca qualunque messaggio in corso.

sa del domodulatore.

I Pin B e C sono collegati fra di loro ed insieme vanno all'uscita del Modem; questa connesione costituirà l'entrata dei segnali per il C64: la ricezione (RX).

Dal Pin M, invece, escono i dati per la trasmissione verso il demodulatore RTTY, il quale provvede a trasformarli da digitali ad analogici, in maniera tale da essere trasmessi via radio.

L'ultimo collegamento è costituito dal Pin E, sul quale è presente una tensione di comando utile per passare dalla ricezione alla trasmissione (PPT) in automatico e va collegato all'ingresso OPERATE del modem.

Tabellina messaggi in TX di F5

1 QRZ
2 Nome e QTH
3 Condizioni di lavoro (messaggio 1)
4 Condizioni di lavoro (messaggio 2)
5 Condizioni di lavoro (messaggio 3)
6 Condizioni di lavoro (messaggio 4)
7 Fine messaggio con convenevoli
8 RY RY RY
9 THE QUICK BROWN etc.

```
1000 GOTO 1100
1010 *************************
1020 *                       *
1030 * RTTY MANAGEMENT  SYSTEM *
1040 *          DI           *
1050 * AMBROGIO MAININI IW1BBG*
1060 *          E            *
1070 *  PAOLO DENARO I2YNA   *
1080 *                       *
1090 *************************
1100 OPEN 1,2,0, CHR$(97)+ CHR$(0)
1110 POKE 666,84
1120 SP=45:CR$= CHR$(13)
1130 IS=1:GS=1
1140 PRINT "[CLEAR]":GOTO 4280
1150 REM                    ASCII
1160 GET T$:IF T$<>""GOTO 1440
1170 GET #1,K$:IF  LEN(K$)=0GOTO 1
     160
1180 IF FZ THEN RETURN
1190 IF LP= PEEK(668)GOTO 1160
1200 IF  POS(0)=40 AND K$=CR$GOTO
     1160
1210 IF  PEEK(214))=19 THEN  SYSP3
     :POKE 214,17:PRINT
1220 IF K$= CHR$(34) THEN POKE 212
     ,0
1230 IF  ASC(K$))31 AND  ASC(K$)<9
     1 OR  ASC(K$)=13 THEN SV%=1
1240 IF SV% THEN PRINT "[GIALLO]"K
     $;:LP= PEEK(668):SV%=0
1250 IF K$= CHR$(34) THEN POKE 212
     ,0
1260 GOTO 1160
1270 GET T$:IF T$<>""GOTO 1440
1280 GET #1,K$:IF  LEN(K$) THEN K=
     ASC(K$) AND 31
1290 IF FZ THEN RETURN
1300 IF LP= PEEK(668)GOTO 1270
1310 IF  POS(0)=40 AND A$(31-K)=CR
     $ THEN 1270
1320 IF  PEEK(214)=)19 THEN  SYSP3
     :POKE 214,17:PRINT
1330 PRINT "[GIALLO]"A$(31-K);:LP=
     PEEK(668)
1340 IF K<>27GOTO 1270
1350 GET T$:IF T$<>""GOTO 1440
1360 REM
1370 GET #1,K$:IF  LEN(K$) THEN K=
     ASC(K$) AND 31
1380 IF  PEEK(668)=LPGOTO 1350
1390 IF  POS(0)=40 AND A$(31-K+32)
     =CR$ THEN 1370
1400 IF  PEEK(214)=)19 THEN  SYSP3
     :POKE 214,17:PRINT
1410 PRINT "[GIALLO]"A$(31-K+32);:
     LP= PEEK(668)
1420 IF K<>31GOTO 1350
1430 GOTO 1270
1440 REM  GET CHR IN RECEIVE MODE
1450 IF T$= CHR$(34) THEN POKE 212
     ,0
1460 IF  ASC(T$))N3GOTO 1530
1470 IF  ASC(T$)<N3 THEN GOSUB 318
     0
1480 IF  LEN(M1$(SR))=U AND T$= CH
     R$(20) THEN M1$(SR)="":GOTO 1
```

```
     620
1490 IF  LEN(M1$(SR)) AND T$= CHR$
     (20) THEN M1$(SR)= LEFT$(M1$(
     SR), LEN(M1$(SR))-1):GOTO 162
     0
1500 IF  LEN(M1$(SR))<255 THEN M1$
     (SR)=M1$(SR)+T$:GOTO 1620
1510 SR=SR+1:IF SR=11 THEN SR=10:T
     $=">":GOSUB 3180
1520 GOTO 1460
1530 IF T$= CHR$(134)GOTO 1660
1540 IF T$="£" THEN PRINT "[RVS][G
     IALLO]ENTER CQ MSG+F7--";:L=1
     2:GOSUB 2580:M$(12)=U$(12):GO
     TO 1610
1550 IF T$="←"GOTO 3120
1560 IF T$= CHR$(133) THEN K=0:POK
     E P1,10:GOTO 1820
1570 IF T$= CHR$(136) THEN GOSUB 2
     500
1580 IF T$= CHR$(137)GOTO 1640
1590 IF T$= CHR$(139) AND SW%=0 TH
     EN SW%=1:IS=0:GOTO 1660
1600 IF T$= CHR$(139) AND SW%=1 TH
     EN SW%=0:IS=0:GOTO 1660
1610 K=Z
1620 IF SW%GOTO 1170
1630 GOTO 1280
1640 POKE P,Q:FZ=1:CA=CA+1:POKE P1
     ,10:PRINT :PRINT "[RVS][BIANC
     O] CQ ON THE AIR N.";CA
1650 L=12:GOSUB 2360:GOTO 2230
1660 REM              SPEED CHANGE
1670 IF SW%=1 THEN POKE 664,9:K=Z:
     W$="ASCII"
1680 IF SW%=0 THEN POKE 664,6:K=Z:
     W$="BAUDOT"
1690 GOSUB 3630:REM     SPEED MENU
1700 GET S$:IF S$="" THEN 1700
1710 IS= VAL(S$):IF IS<1 OR IS>9GO
     TO 2280
1720 IF IS=1 THEN POKE 665,250:POK
     E 666,84:POKE 661,25:POKE 662
     ,38:SP=45:IS=1
1730 IF IS=2 THEN POKE 665,250:POK
     E 666,76:POKE 661,25:POKE 662
     ,38:SP=50:IS=2
1740 IF IS=3 THEN POKE 665,80: POK
     E 666,51:POKE 661,68:POKE 662
     ,25:SP=75:IS=3
1750 IF IS=4 THEN POKE 665,252:POK
     E 666,34:POKE 661,26:POKE 662
     ,17:POKE 659,99:SP=110
1760 IF IS=5 THEN POKE 665,168:POK
     E 666,25:POKE 661,112:POKE 66
     2,12:POKE 659,5:SP=150
1770 IF IS=6 THEN POKE 665,212:POK
     E 666,12:POKE 661,6:POKE 662,
     6:POKE 659,6:SP=300
1780 IF IS=7 THEN POKE 665,106:POK
     E 666,6:POKE 661,209:POKE 662
     ,2:POKE 659,7:SP=600
1790 IF IS=8 THEN POKE 665,54:POKE
      666,3:POKE 661,55:POKE 662,1
     :POKE 659,8:SP=1200
1800 IF IS=9 THEN POKE 665,154:POK
     E 666,1:POKE 661,105:POKE 662
     ,0:POKE 659,10:SP=2400
1810 GOTO 2280
1820 POKE P,Q:F=Z:G=Z:PRINT :PRINT
      "[7 DOWN][RVS][GIALLO]";SP;"
     [LEFT] BAUD "W$" ON THE AIR"
1830 A%=1904
1840 M$="◤RYRYRYRYRYRYRYRYRY"+M$(1
     4)+"◤":GOSUB 2380
1850 IF  LEN(M1$(Z)) THEN  GOSUB 3
     290
1860 GET T$:IF T$="" THEN GOSUB 21
     60:GOTO 1860
1870 IF T$= CHR$(34) THEN POKE 212
     ,0
1880 IF SU%=1 AND T$= CHR$(13) THE
     N T$="◤":GOTO 3350
1890 IF SU%=1 AND T$= CHR$(20) THE
     N GOSUB 3460:GOTO 1860
1900 L= ASC(T$):IF (L<T1 OR L>N3)
     AND SU%=1 AND  LEN(H$)>0 THEN
      3390
1910 L= ASC(T$):IF (L<T1 OR L>N3)G
     OTO 2790
1920 IF SU%=1 THEN 3350
1930 GOSUB 1940:GOTO 1860
1940 I= ASC(T$):IF (I<T1 OR I>N3)
     THEN RETURN
1950 IF RT%=0 THEN PRINT  "[RVS][G
     IALLO]";T$;
1960 IF T$= CHR$(34) THEN POKE 212
     ,0
1970 IF SW%=1 THEN 2010
1980 I=B%(I-T):G=Z:IF I>T THEN I=I
     -T:G=U
1990 IF F=Z AND G=U THEN GOSUB 213
```

```
    0
2000 IF F=U AND G=Z THEN GOSUB 216
     0
2010 AB=AB+U:IF AB>ML AND T$=" " T
     HEN GOSUB 2190:RETURN
2020 IF SW%=1 THEN PRINT#1,T$;
2030 IF SW%=0 THEN PRINT#1, CHR$(C
     %(I));
2040 IF  PEEK(669)<> PEEK(670) THE
     N 2040
2050 RETURN
2060 REM                SYNC. PRINT
2070 IF SW%=1 AND RT%=0 AND (T$=CR
     $ OR T$="◤") THEN PRINT#1,CR$
     ;:PRINT :GOTO 2110
2080 IF SW%=1 AND RT%=1 AND (T$=CR
     $ OR T$="◤") THEN PRINT#1,CR$
     ;:GOTO 2110
2090 IF SW%=1 THEN PRINT#1, CHR$(2
     55);
2100 IF SW%=0 THEN PRINT#1, CHR$(3
     1-S);
2110 IF  PEEK(669)<> PEEK(670)GOTO
      2110
2120 RETURN
2130 REM                       FIGS
2140 IF SW% THEN RETURN
2150 S=4:F=U:GOSUB 2060:RETURN
2160 REM                       LTRS
2170 IF SW%=1 THEN T$="":GOTO 2060
2180 S=Z:F=Z:GOSUB 2060:RETURN
2190 REM                      CR+LF
2200 IF RT%=1 AND SU%=0 THEN AB=Z:
     RETURN
2210 IF SW%=1 THEN AB=0:PRINT :PRI
     NT#1, CHR$(13);:RETURN
2220 S=23:GOSUB 2060:S=29:GOSUB 20
     60:GOSUB 2160:PRINT  CHR$(13)
     ;:AB=Z:RETURN
2230 REM                     TX END
2240 L=20:GOSUB 2360
2250 IF  PEEK(669)<> PEEK(670) THE
     N 2250
2260 POKE P,Z:IF FZ=U THEN GOSUB 3
     520
2270 IF FZ=1GOTO 1640
2280 POKE P1, 5:PRINT :PRINT "[5 R
     IGHT][RVS][VERDE]";SP;"[LEFT]
      BAUD RX "W$" [DOWN][BIANCO]"
     :PRINT SP$;SP$;SP$;SP$:PRINT
2290 FOR J=1 TO 255:GET #1,K$:NEXT
2300 K=Z:POKE P,Z:PRINT  CHR$(142)
     ;:GOTO 1620
2310 REM                OUT FIX MSG
2320 GET T$:IF T$="" THEN GOSUB 21
     60:GOTO 2320
2330 IF T$= CHR$(34) THEN POKE 212
     ,0
2340 IF T$= CHR$(135) THEN L=29:GO
     TO 2360
2350 L= VAL(T$):PRINT "[RVS][BIANC
     O]";:IF L<0 OR L>9 THEN RETUR
     N
2360 M$=M$(L):IF M$="" THEN RETURN
2370 REM               OUT FIX MSG.
2380 FOR M=U TO  LEN(M$):GET B$:IF
      B$=""GOTO 2410
2390 IF T$= CHR$(34) THEN POKE 212
     ,0
2400 M$(25)=M$(25)+B$:IF B$= CHR$(
     V) THEN M$=""
2410 T$= MID$(M$,M,U):IF T$=""GOTO
      2440
2420 IF  ASC(T$)<T1 OR  ASC(T$)>N3
     GOTO 2460
2430 GOSUB 1940
2440 NEXT M:M$=M$(25):M$(25)="":IF
      M$="" THEN RETURN
2450 GOTO 2380
2460 IF T$="◤" OR T$=CR$ THEN GOSU
     B 2200:GOSUB 2160
2470 IF T$= CHR$(135) THEN M=M+U:L
     = VAL( MID$(M$,M,U)):M$=M$(L)
     :GOTO 2380
2480 IF T$="↑" THEN L=14:GOTO 2360
2490 GOTO 2440
2500 REM                  INPUT MSG
2510  SYSP3:POKE 214,17:PRINT
2520 POKE P,Z:PRINT "[RVS][GIALLO]
     MSG CODE?";
2530 GET T$:IF T$=""GOTO 2530
2540 IF T$= CHR$(34) THEN POKE 212
     ,0
2550 IF  PEEK(214)=>19 THEN  SYSP3
     :POKE 214,17:PRINT
2560 L= VAL(T$)
2570 PRINT "[RVS]";L;" OK";:PRINT
     "[RVS][GIALLO]";
2580 U$(L)=""
2590 GET T$:IF T$=""GOTO 2590
2600 IF  PEEK(214)=>19 THEN  SYSP3
     :POKE 214,17:PRINT
```

```
2610 IF T$= CHR$(34) THEN POKE 212
     ,0
2620 IF T$= CHR$(136) OR T$="←" TH
     EN PRINT "[GIALLO]OK[RVOFF]";
     :IF U$="" THEN RETURN
2630 U$(L)=U$(L)+T$:FS=1:PRINT "[G
     IALLO]";T$;:GOTO 2590
2640 REM                OUT MSG
2650 GET T$:IF T$="" THEN GOSUB 21
     60:GOTO 2650
2660 IF T$= CHR$(34) THEN POKE 212
     ,0
2670 L= VAL(T$)
2680 IF U$(L)="" THEN RETURN
2690 M$=U$(L):GOSUB 2380
2700 RETURN
2710 REM           CW ID. FOLLOW
2720 M$=" PSE K K K▼CW IDENTIFY FO
     LLOW":GOSUB 2380
2730 EX=65:POKE P,B6:FOR J=1 TO 80
     0:NEXT
2740 FOR I=U TO  LEN(CW$):P$= MID$
     (CW$,I,U)
2750 IF P$="P" THEN POKE P,BO:POKE
      S3,232:FOR J=U TO EX:NEXT :P
     OKE P,B6:POKE S3,Z
2760 IF P$="L" THEN POKE P,BO:POKE
      S3,232:FOR J=U TO EX*3:NEXT
     :POKE P,B6:POKE S3,Z
2770 IF P$=" " THEN FOR J=U TO EX:
     NEXT
2780 FOR J=U TO EX:NEXT :NEXT I:FO
     R J=U TO 800:NEXT :GOSUB 2200
     :GOTO 2260
2790 REM               TX CNTRL
2800 IF T$= CHR$(138) THEN M$=" -
     "+ LEFT$(TI$,2)+":"+ MID$(TI$
     ,3,2)+" GMT -"+GG$:GOSUB 2380
2810 IF T$= CHR$(13) THEN GOSUB 22
     00:GOTO 1860
2820 IF T$= CHR$(19) THEN GOSUB 22
     00:GOTO 2260
2830 IF T$="↑" THEN L=14:GOSUB 236
     0:GOTO 1860
2840 IF T$="←" THEN PRINT "[RVOFF]
     ([RVS]";:M$(14)="":GOTO 3070
2850 IF T$="£" THEN L=11:GOSUB 236
     0:GOTO 2260
2860 IF T$= CHR$(136) THEN GOSUB 2
     640:GOTO 1860
2870 IF T$= CHR$(133)GOTO 2230
2880 IF T$= CHR$(135) THEN GOSUB 2
     310:IF L=7GOTO 2230
2890 IF T$= CHR$(139)GOTO 2710
2900 IF T$= CHR$(134) AND SU%=1 TH
     EN SU%=0:RT%=0:PRINT "▚";:GOT
     O 2920
2910 IF T$= CHR$(134) AND SU%=0 TH
     EN SU%=1:RT%=0:PRINT "[RVOFF]
     ♠";
2920 IF T$<> CHR$(140)  THEN  1860
2930 RT%=1
2940 FOR J=1024  TO  1823 STEP 40
2950 FOR JJ=0 TO 39
2960 R%= PEEK(J+JJ) AND 127:POKE J
     +JJ,R%+128
2970 POKE MN+J+JJ,1
2980 IF R%<32 THEN R%=R%+64
2990 T$= CHR$(R%):GOSUB  1940
3000 NEXT
3010 IF SW%=0 THEN S=23:GOSUB 2060
     :S=29:GOSUB 2060:GOSUB 2160:A
     B=Z
3020 IF SW%=1 THEN T$="▼":GOSUB 20
     60:AB=Z
3030 GET T$:IF T$="[HOME]" THEN J=
     2000
3040 NEXT
3050 RT%=0
3060 GOTO 1860
3070 REM                   OM CALL
3080 GET T$:IF T$=""GOTO 3080
3090 IF T$= CHR$(34) THEN T$="'"
3100 IF T$="←" THEN PRINT "[RVOFF]
     )[RVS]";:M$(14)=" "+M$(14):GO
     TO 1860
3110 GOSUB 2580:M$(14)=M$(14)+T$:G
     OTO 3080
3120  SYSP3:POKE 214,17:PRINT
3130 L=14:PRINT "[GIALLO][RVS]CALL
      SIGN?";
3140 GOSUB 2580:IF U$(14)="" THEN
     FS=Z
3150 M$(14)=" "+U$(14)+" "+M$(0)
3160 M$(7)=M$(14)+M$(17)
3170 GOTO 1620
3180 REM  ACQUISIZIONE CARATTERI
3190 REM
3200 B%= ASC(T$):IF B%>64 THEN B%=
     B%-64
3210 IF T$= CHR$(20) THEN POKE A%,
     32:A%=A%-1:B%=32
```

```
3220 IF T$= CHR$(13) THEN B%=105
3230 IF A%(1904 THEN A%=1904
3240 POKE A%,B%:POKE MN+A%,7
3250 IF T$= CHR$(20) THEN A%=A%-1
3260 A%=A%+1:IF A%)2022 THEN A%=20
     22: SYSP4
3270 POKE A%,82:POKE MN+A%,7
3280 RETURN
3290 REM              OUT MEM. MSG.
3300 FOR SR=0 TO 10
3310 M$=M1$(SR):IF M$="" THEN 3330
3320 GOSUB 2380:M1$(SR)=""
3330 NEXT SR
3340 SR=0:RETURN
3350 REM               WORD TX MODE
3360 H$=H$+T$:PRINT T$;
3370 IF T$=" " OR T$="▼" THEN 3390
3380 GOTO 1860
3390 OP= LEN(H$):FOR J=U TO OP
3400 T$= MID$(H$,J,1)
3410 PRINT  LEFT$(RT$,OP-J+1);"[GI
     ALLO][RVS]";:IF T$="▼" AND SW
     %=1 THEN GOSUB 2060
3420 IF T$="▼" AND SW%=0 THEN GOSU
     B 2190
3430 GOSUB 1940:PRINT  "[RVOFF]";
     LEFT$(RU$,OP-J);
3440 NEXT J
3450 H$="":GOTO 1860
3460 REM               DELETE CHAR.
3470 IF  LEN(H$)=0 THEN RETURN
3480 PRINT T$;
3490 IF  LEN(H$)=1 THEN H$="":RETU
     RN
3500 H$= LEFT$(H$, LEN(H$)-1)
3510 RETURN
3520 REM      CHECK RY IN CQ MODE
3530 FOR J=U TO 255:GET #1,T$:NEXT
      :PK=TI/60:TF=0
3540 IF TI/60)PK+15 THEN RETURN
3550 GOSUB 1620:IF SW%=0 AND A$(31
     -K)()"R" THEN 3610
3560 IF SW%=1 AND K$()"R" THEN 361
     0
3570 GOSUB 1620:IF SW%=0 AND A$(31
     -K)()"Y" THEN 3540
3580 IF SW%=1 AND K$()"Y" THEN 354
     0
3590 IF TF=0 THEN TF=1:GOTO 3610
3600 FZ=Z:CA=Z:RETURN
3610 GET T$:IF T$()"[HOME]" THEN 3
```

```
     540
3620 GOTO 3600
3630 PRINT "[CLEAR]    VELOCITA' D
     ISPONIBILI ";W$
3640 PRINT " (1) = 45.45 BAUD"
3650 PRINT " (2) =    50 BAUD"
3660 PRINT " (3) =    75 BAUD"
3670 PRINT " (4) =   110 BAUD"
3680 PRINT " (5) =   150 BAUD"
3690 PRINT " (6) =   300 BAUD"
3700 PRINT " (7) =   600 BAUD"
3710 PRINT " (8) =  1200 BAUD"
3720 PRINT " (9) =  2400 BAUD"
3730 PRINT " (0) =  INVARIATO"
3740 RETURN
3750 N$=" I2YNA "
3760 DIM M$(30)
3770 M$(0)=" DE"+N$
3780 M$="▼CQ CQ CQ CQ"+M$(0)
3790 M$(12)=M$+M$+M$+M$
3800 M$(9)="▼THE QUICK BROWN FOX J
     UMPS OVER THE LAZY DOG'S BACK
      0123456789 DE"
3810 M$(9)=M$(9)+N$
3820 M$(8)="▼RYRYRYRYRYRYRYRYRYRYR
     YRYRYRYRYRYRYRYRYRYRYRYRYRYRY
     RYRYRYRYRY▼"
3830 M$(20)="▼PSE K K K▼"
3840 REM
3850 M$(2)="▼IL MIO NOME E' PAOLO
     - QTH COLOGNO -MI-▼QRA LOCATO
     R - EF 37 F -▼"
3860 M$(3)="▼RTX YAESU FT-480/R▼AN
     TENNA PKW 9 EL. VERT.▼"
3870 M$(3)=M$(3)+"RTTY CBM 64▼MODE
     M TU 170/V▼"
3880 M$(1)="▼RYRYRYRY QRZ? QRZ? QR
     Z?"+M$(0)+"←"
3890 M$(4)="▼RTX ICOM IC-720/A▼ANT
     . PKW 12 ELEM. LOG▼RTTY CBM 6
     4 MODEM TU 170/V▼"
3900 M$(5)=""
3910 M$(6)=""
3920 M$(11)="▼▼000000     0000▼0
      0  0    0  CIAO▼     0
      0  CIAO▼"
3930 M$(11)=M$(11)+"    0        000
        DE▼  0            0 "+N$+"▼
      0      0      0 "
3940 M$(11)=M$(11)+" ITALIA▼"
3950 M$(11)=M$(11)+"   0       0000▼
```

```
     ◤SK SK SK◤←"
3960 M$(29)="◤RYRYRYRYRYRYRYRYRYRY
     RYRYRYRYRYRYRYRYRYRY◤"
3970 IF GS THEN INPUT "GMT - (HHMM
     ) ";TA$:IF  LEN(TA$)<>4 THEN
     3990
3980 TI$=TA$+"00"
3990 IF GS THEN INPUT "DATA       ---
     ) ";GG$
4000 M$(14)=N$
4010 SP$="
                            "
4020 RETURN
4030 P1=53280:POKE P1,8:PRINT "[BI
     ANCO]"; CHR$(14);:P2=56577:PO
     KE P2,Z
4040 DIM A$(63):FOR K=0 TO 63:READ
      A$(K):NEXT
4050 DATA "",V,X,M,"",G,B,O,Q,P,Y,
     H,W,L,Z,T,K,C,F,N,J,R,D,"",U,
     I,S," ",A,"";E,""
4060 DATA "",;,/,.,"",&,?,9,1,0,6,
     #,2,),+,5,(,":","!",",",',4,$
     ,"",7,8
4070 DATA *," ",-,"",3,""
4080 A$(29)= CHR$(13):A$(61)= CHR$
     (13)
4090 Z=0:U=1:D=2:Q=4:V=20:T=32:W=1
     7:N=90:NR=2:DIM U$(20):VE=20
4100 P=56577:A%=1904
4110 RT$="[39 LEFT]"
4120 DIM C%(32)
4130 RU$="[39 RIGHT]"
4140 DIM B%(63):FOR I=Z TO 63:READ
      B%(I):NEXT
4150 DATA 27,41,46,58,45,31,52,37,
     33,54,43,46,57,39,56,40,50,34
     ,38,47,53,62,42,35
4160 DATA 51,60,49,48,31,31,31,44,
     31,7,12,17,13,15,9,20,26,19,5
     ,1,22,24,25,28,18,2
4170 DATA 21,11,30,3,16,6,8,10,14,
     31,31,31,31,31
4180 FOR I=Z TO 31:READ C%(I):NEXT

4190 B0=20:B6=Q:VV=53281
4200 POKE VV,11:TE$="18"
4210 UK=218:SN=36877:P3=49152
4220 DATA 31,15,23,7,31,11,19,3,29
     ,13,21
4230 DATA 5,25,9,17,1,30,14,22,6,2
     6,10,18
4240 DATA 31,28,12,20,4,24,31,16,3
     1,31
4250 POKE 650,128:POKE 649,15:ML=6
     2:CW$="PP PLL PLLLL PL PPLP"
4260 W$="-BAUDOT-":IF SW%=1 THEN W
     $="-ASCII-"
4270 MN=54272:NR=1:T1=31:N3=91:GOS
     UB 3750:PRINT "[20 DOWN]":GOT
     O 2280
4280 REM                 LM CHARGE
4290 J=49152:P3=49152:P4=49408
4300 READ A$:IF A$<>"***" THEN 430
     0
4310 READ A$:IF  A$="****" THEN
     RESTORE :GOTO 4030
4320 IF A$="**" THEN J=P4:GOTO 431
     0
4330 POKE J, VAL(A$)
4340 J=J+1
4350 GOTO  4310
4360 DATA  ***,162,7,169,216,141,3
     7,192,169,4,234,141
4370 DATA 34,192,141,31,192,169,40
     ,141,30,192,169
4380 DATA 0,141,33,192,141,36,192,
     173,0,0,141
4390 DATA 0,0,142,0,0,238,36,192,2
     08,3,238
4400 DATA 37,192,238,33,192,208,3,
     238,34,192,238
4410 DATA 30,192,208,3,238,31,192,
     173,31,192,201
4420 DATA 7,208,8,173,30,192,201,7
     2,208,1,96
4430 DATA 76,29,192,255,**
4440 DATA 162,1,169,219,141,37,193
     ,169,7,234,141
4450 DATA 34,193,141,31,193,169,11
     3,141,30,193,169
4460 DATA 112,141,33,193,141,36,19
     3,173,231,7,141
4470 DATA 191,7,142,191,219,238,36
     ,193,208,3,238
4480 DATA 37,193,238,33,193,208,3,
     238,34,193,238
4490 DATA 30,193,208,3,238,31,193,
     173,31,193,201
4500 DATA 7,208,8,173,30,193,201,2
     32,208,1,96
4510 DATA 76,29,193,****
```

COME PROTEGGERSI DAGLI SPROTETTORI

di **Ernesto Sidoti**

Crei programmi per il Commodore?

E senza dubbio ti sarai posto il problema di proteggere il tuo software e magari nessuna idea ti sarà venuta in mente. Ho indovinato? Allora leggi questo mio articolo, sicuramente ti sarà d'aiuto.

Bando ai preamboli. Batti subito la riga:

```
10 PRINT"CIAO":REM"TTTTTTTTTTTTT
```

Le T si creano così:

Dopo aver battuto: REM devi premere i tasti in quest'ordine:

1) digita 2 doppi apici, le così dette virgolette ("), prorpio due volte!

2) con il tasto (INST/DEL) cancellare un doppio apice;

3) premere contemporaneamente (SHIFT) (INST/DEL) per circa 2 secondi;

4) premere il tasto (INST/DEL) un numero di volte pari ai caratteri da cancellare (nel caso dell'esempio: 19 volte)

Ora, se dai il comando RUN, il programma costituito dalla sola riga 10, visualizzerà: "CIAO" Ma quando vorrai ottenere il LISTato non apparirà. In questo modo puoi inserire righe non visualizzabili nel tuo programma.

Scrivi adesso il programma di prova a corredo

Adesso salva il programma in questo modo:
SAVE "PROVAAT"

Come ben saprai nel tuo Commodore 64 esiste un buffer per il TAPE (il buffer non è altro che un piccolo magazzino di caratteri).

Questo buffer risiede tra la locazione 828 e la locazione 1019.

Al momento della SAVE nella locazione 833 si troverà il valore numerico corrispondente, in codice ASCII, alla prima lettera del nome del programma; nella locazione 834 il valore della seconda lettera, nella 835 il valore della terza e così via.

In questo caso, avendo salvato il programma con il nome PROVAAT, nella locazione 833 si troverà il valore 80 (in codice ASCII "P"), nella successiva leggeremo 82, poi 79, ecc.

In fine, nella locazione 839 troveremo il valore 20 corrispondente alla T, infatti quella "T" che è contenuta nel nome del programma non è il carattere "t" dell'alfabeto, ma vi apparirà in campo inverso e starà a significare l'azione di cancellazione di un carattere.

A questo punto dovresti aver capito il trucco.

Salvo il programma con il nome PROVAAT, in tal modo nella locazione 839 sarà memorizzato il valore 20 . Così facendo quando carico il programma

```
5 PRINT"(CLR/HOME)EQUAZIONI
  DI II GRADO":PRINT
10 PRINT"SCRIVI I VALORI DI A,B,C"
20 INPUT A,B,C:S=PEEK(839)
  :REM"TTTTTTTTTTTTTTTTTTT
30 PRINT:IFS<>20THENSYS64732
  :REM"TTTTTTTTTTTTTTTTTTTTTTTTTTT
40 D=(B 2-4(A*C))
45 IFD<OTHEN PRINT"(CLR/HOME)
   DELTA NEGATIVO":END
48 R=SQR(D)
50 X1=(-B+R)/(2*A)
60 X2=(-B-R)/(2*A)
70 PRINT
80 PRINT"A=";A,"B=";B,"C=";C
85 PRINT
90 PRINT"        LE RADICI SONO:"
92 PRINT"X1=";X1,"X2=";X1
95 END
```

originale ritroverò nella locazione 839 sempre il valore 20.

Vogliono copiarti il programma?

Prima di tutto dovranno caricarlo nella memoria del Commodore con un normale procedura "LOAD" (RETURN), quindi appena trovato il programma il calcolatore visualizzerà:

FOUND PROVA, non "PROVAAT"

Gli ultimi due caratteri non vengono visualizzati. O meglio, vengono sì visualizzati, ma l'ultima "A" viene subito cancellata dal comando "T", e l'occhio non se ne avvede.

Il "pirata" che voglia duplicare il programma scriverà:

SAVE "PROVA" o solo SAVE.

Così facendo, nel primo caso, metterà nelle locazioni riservate al nome del programma solo i valori delle lettere "P, R, O, V, A"; nel secondo caso, invece, tutte le locazioni riservate al nome avranno valore 32, equivalente allo spazio.

A questo punto quando il calcolatore eseguirà ciò che è stato definito nella riga 30 e .... 0, non trovando ciò che dovrebbe esistere nel famoso buffer, eseguirà il comando, ormai noto di system reset, la SYS (64738).

Se ancora non sei soddisfatto, batti come prima riga del programma la seguente:

1 REMERNESTO

Adesso esegui questi comandi diretti:

1) FOR A = 2054 TO 2059: POKEA, 20: next (RETURN)

2) POKE 2060, 204 (RETURN)

Battendo comando LIST avrai solamente un bel: "SYNTAX ERROR".

Perchè tutto ciò?

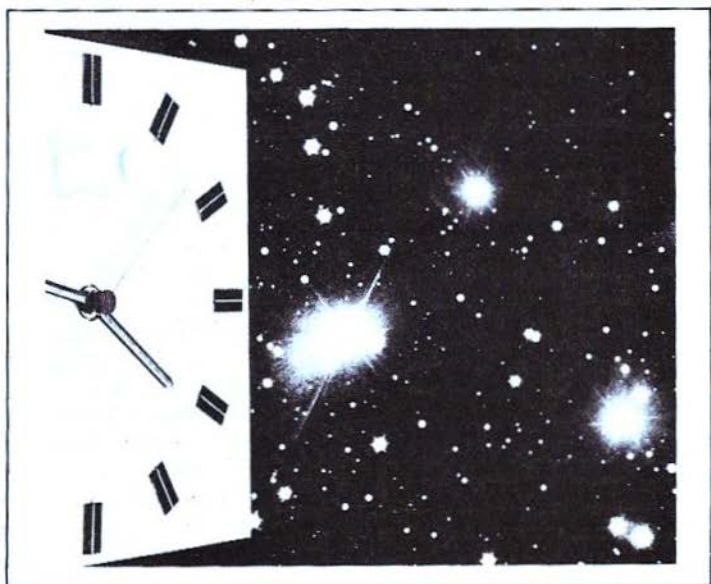
Nella fase 1), come l'iterazine descritta, vengono sostituite le lettere del nome "ERNESTO" con 7 "T", corrispondenti a sette volte il dasto DEL.

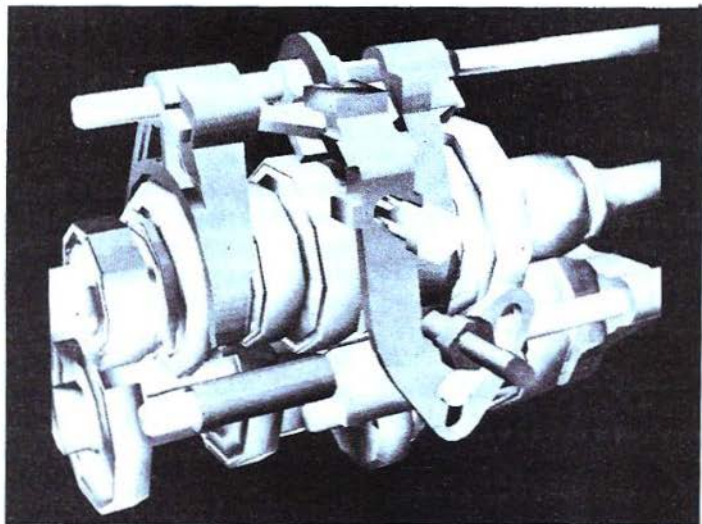
Nella fase 2) nella locazione 2060 viene scritto un valore non compreso nel codice di comandi; quindi quando l'interprete BASIC andrà a leggere tale valore dovrà interpretarlo come codice di comando e segnalerà un errore di sintassi.

Un'altra protezione di notevole interesse riguarda la locazione di memoria numero 808.

Porvate, infatti, ad eseguire la POKE 808, 205. Con questo sistema cambio l'indirizzo di partenza del vettore che controlla la sobroutine che analizza se il tasto (RUN/STOP) è premuto oppure no. In questa maniera disabilito non solo il tasto (RUN/STOP) ‹ RESTONE › ma anche l'istruzione LIST.

un'altra locazione interessante è la 819, infatti con POKE 819, PEEK (817) metto nel vettore di SAVE l'indirizzo della routine di LOAD rendendo quindi inutilizzabile la prima.

Avete capito bene; Allora, mi raccomando: silenzio!

NUMERI PRIMI

di **Eugenio Coppari**

Alcuni rami della teoria dei numeri sono in grado di suscitare un notevole interesse anche nelle persone che non si occupano abitualmente di matematica. In questi articolo affronteremo il problema dei numeri primi, ponendo in particolare rilievo gli aspetti più curiosi e interessanti concernenti questo insieme numerico.

Prima di iniziare ad analizzare questo argomento è necessario ricordare la definizione di numero primo. Un numero naturale si dice "primo" quando è divisibile solo per se stesso e per l'unità, ad esempio 5,13,29 sono numeri primi. Come si può facilmente capire, il concetto di numero primo è particolarmente facile ed intuitivo, poichè non presenta sostanziali difficoltà che ne rendano ardua la comprensione.

Questa facilità di assimilazione del concetto non dovrà peraltro farci ritenere questo argomento banale, infatti pochi altri campi della matematica sono stati oggetto di analisi così approfondita come quello dei numeri primi.

Sin dall'antichità questo insieme numerico è stato al centro delle analisi dei matematici di ogni tempo. Dapprima Euclide scoprì che esso è infinito e ciò fu possibile dimostrando che la negazione di tale asserzione conduce a contraddizioni di carattere logico. In matematica questo tipo di dimostrazione è detta "per assurdo" e viene usata assai frequentemente".

Dopo questa prima importante scoperta, si pose il problema di determinare la densità di questo insieme all'interno di quello dei numeri naturali, la soluzione di questo enigma è giunta in tempi relativamente recenti ed è espressa dal seguente limite:

$$\lim_{n \to \infty} \frac{n}{\log(n)}$$

infatti, se noi poniamo uguale a valori sempre più alti, otterremo risultati di densità che si avvicinano sempre maggiormente al valore cercato.

I due teoremi citati in precedenza, il secondo dei quali prende il nome dei "teorema dei numeri primi" data la sua enorme importanza, costituiscono due capisaldi di questo ramo della matematica.

Essere riusciti a risolvere questi enigmi è senza dubbio importante, ma altri problemi, fino ad oggi privi di risposta e che quindi continuano ad essere di profondo interesse da parte di matematici.

Uno di essi spicca rispetto agli altri: esiste una logica, esprimibile magari attraverso una determinata relazione matematica, che permetta di comprendere la maniera in cui la sequenza dei numeri primi si dispone all'interno dell'insieme dei numeri naturali?

Sino ad ora si sa solamente, attraverso un importante teorema, che non esiste alcuna formula polinominale che permetta di generare tutti i numeri primi uno diverso dell'altro in maniera tale che nessun elemento primo rimanga escluso o ripetuto più volte.

Diamo ora un breve esempio di quanto appena asserito.
L'espressione X^2, + X + 17 costituisce la più nota formula generatrice di numeri primi, essa è stata creata la Eulero e fornisce valori esatti per $0<x<15$, dopodichè è del tutto insufficiente per il compito che le è stato affidato.

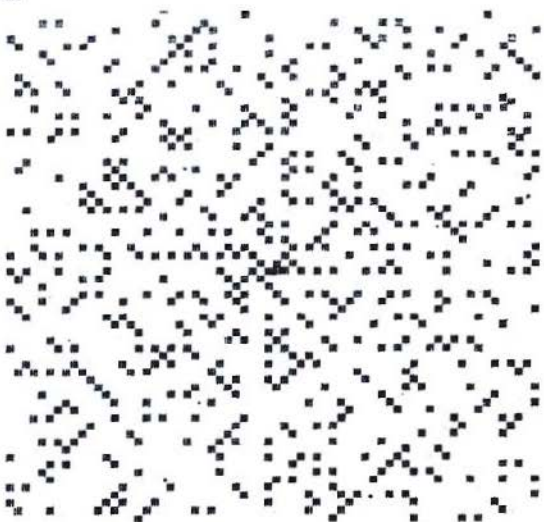
Ma allora quale può essere la logica che guida la disposizione dei numeri primi, se fino ad ora le risposte fornite si sono rilèvate parziali o errate?

Forse fondendo razionalità, immaginazione e fantasia potremo meglio addentrarci in questa città, quella dei numeri primi, che purtroppo sembra non possedere una lista completa degli indirizzi dove reperire i suoi abitanti.

Come spesso accade in quasi tutti le discipline scientifiche, alcuni importanti scoperte si verificano in circostanze inusuali e difficilmente preventivabili, basterebbe a questo proposito ricordare che Fermi realizzò la sua prima reazione nucleare all'interno della vasca di pesci presente nella facoltà di fisica di Roma.

Ho citato questa situazione in quanto qualcosa di analogo è avvennuto nel ramo matematico che ci interessa in questo momento.... Stanislaw Ulam, ricercatore del Los Alamos Laboratory, in una giornata dell'anno 1963, stava seguendo un convegno scientifico a cui era stato invitato a partecipare.

I lavori del seminario si sviluppavano in maniera particolarmante noiosa senza richiamare l'interesse del nostro personaggio che era più propenso a viaggiare con la sua mente verso altri lidi.

Dopo aver scarabocchiato alcuni numeri su un blocchetto di carta decise di disporre i numeri naturali su una spirale in modo che, partendo dal centro, si allargasse in progressione seguendo questa configurazione. Il suo stupore fu subito grande: i numeri primi all'interno di questa sequenza si disponevano in maniera tutt'altro che casuale. Dapprima pensò che ciò fosse da addebitare al fatto che gli elementi presi in onsiderazione sul suo notex erano troppo e quindi certe simmetrie e disposizioni regolari dei numeri primi sarebbero scomparse procedendo ad analizzare successioni sempre più grandi di elementi.

La spirale dei numeri primi prodotta con quadrati da 9 punti (tempo di visualizzazione: 25 minuti).

Incuriosito si ripropose, terminato il convegno, di esaminare sequenze sempre più lunghe di numeri naturali disposti su una spirale, al fine di osservare il comportamento degli elementi primi.

Avvalendosi dell'aiuto di un calcolatore molto potente e di un disco contenente i primi 90 milioni di numeri primi, scoprì che quello che aveva notato sul suo blocco di carta persisteva: gli elementi primi tendevano a disporsi lungo linee rette in quasi tutte le zone della spirale, accentuando naturalmente questo comportamento in determintate aree la loro presenza era maggiore.

Queste linee rette, lungo le quali tendono a disporsi i numeri primi, sono di ogni genere: orizzontali, verticali,. oblique e contengono elementi che sono i valori corrispondenti di equazioni quadratiche, cioè di secondo grado, aventi come primo elemento 4 x elevato al quadrato. Ad esempio la sequenza diagonale dei numeri primi 5, 19, 41, 71 è data dalla relazione 4x elevato al quadrato +2 x +17, dove x assume i valori compresi tra 0 e 3.

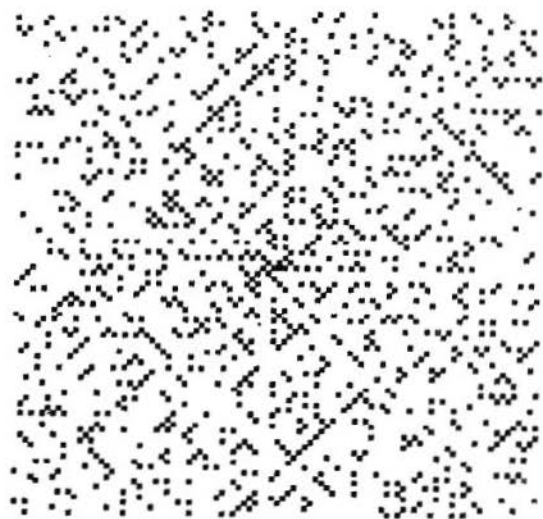
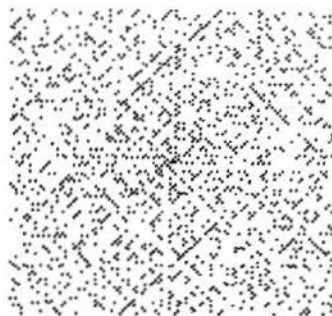
Naturalmente è possibile iniziare il processo di spiralizzazione dei numeri senza cominciare da 1. Se infatti volessimo scegliere 17 come primo elemento otterremmo che l'insieme dei numeri primi, che si trova sulla diagonale che è condotta tra il vertice in alto a destra e il corrispettivo in basso a sinistra, è espresso dalla relazione x elevato al quadrato +x+17.

É quindi chiaro che si possono ottenere altre interessanti configurazioni di numeri primi semplicemente mutando la locazione di partenza.

Un'analisi così interessante come quella condotta da Ulam conduce, come daltronde accade spesso in matematica, a nuove congetture e problemi. Ci potremmo chiedere, ad esempio, se esistono linee rette conteneti infiniti elementi, in quale varia la densità dei numeri primi dalle zone inferioni a quelle superiori della diagonale e ancora molti altri questiti che uno studio assiduo ci potrebbe proporre.

Purtroppo dobbiamo ancor oggi riconoscere che la patina di fascino e di mistero che avvolge questi numeri è ben lungi dall'essere tolta nonostante i notevoli sforzi condotti dall'uomo con l'ausilio dei mezzi tecnologici.

Ancor oggi un numero con la base di una cifra e l'esponente di 5 rappresenta il limite quantitativo alle nostre conoscenze su questa classe numerica: (2 elevato a 19937)-1.

In questo articolo si sono citate solo una parte delle "curiosità" concernenti i numeri primi. Potremmo ricordare, ad esempio i numeri di Marsenne dove la realazione 2 elevato alla (P-1), al variare di p, pro-

La spirale dei numeri primi prodotta con quadratini da 4 punti (tempo di visualizzazione: un'ora e 1/2 circa).

La spirale dei numeri primi prodotta punto a punto (tempo di visualizzazione: ore 3 e 30 minuti).

duce una notevole quantità di elementi primi. Ma il nostro intento era soprattutto quello di sottolineare quegli aspetti che più facilmente potremmo avere un contatto con il calcolatore e con le sue capacità grafiche.

In conseguenza di quanto è stato detto, proponiamo il seguente programma che affronta i temi evidenziati nell'articolo.

Il programma consente di visualizzare le configurazioni di numeri primi locate rispettivamente su una spirale, su un triagolo e su un esagono, a seconda della opzione che si è scelta.
All'interno di ciascuna opzione è possibile scegliere la dimensione dell'elemento da mettere in rilievo, naturalmente tanto minore sarà la dimensione di quest'ultimo e tanto maggiori saranno le nostre analisi quantitative.

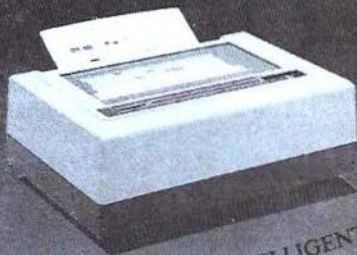
Se, ad esempio, abbiamo scelto una configurazione a spirale e una risoluzione di 2x2 dot, l'elemento verrà disegnato se è un numero primo altrimenti no.

Non riteniamo necessario soffermarci ulteriormente nella descrizione del programma, poichè è sufficientemente autoesplicano comprensibile per chi ne vuole fare uso. Coloro che ritengono di poter fornire un contributo originale sono pregati di far pervenire il materiale alla redazione della rivista che lo vaglierà ed eventualmente lo pubblicherà.

```
1000 REM ************************
1010 REM *                      *
1020 REM * IL MISTERO DEI       *
1030 REM *                      *
1040 REM *       NUMERI  PRIMI  *
1050 REM *                      *
1060 REM *                      *
1070 REM * DI EUGENIO COPPARI   *
1080 REM *                      *
1090 REM ************************
1100 :
1110 DIM CL(17)
1120 PRINT "[CLEAR]"; TAB(7);"[6 D
     OWN][BIANCO][RVS]
                          "
1130 PRINT  TAB(7);"[BIANCO][RVS]
     DISPOSIZIONI PARTICOLARI "
1140 PRINT  TAB(7);"[BIANCO][RVS]
                                 "
1150 PRINT  TAB(7);"[BIANCO][RVS]
        DEI NUMERI PRIMI         "
1160 PRINT  TAB(7);"[BIANCO][RVS]
                                 "
1170 PRINT  TAB(7);"[BIANCO][RVS]
        (ATTENDERE PREGO)        "
1180 PRINT  TAB(7);"[BIANCO][RVS]
                                 "
1190 FOR K=1 TO 4000:NEXT :PRINT "
     [CLEAR]"
1200 PRINT  TAB(11);"[BIANCO][RVS]
        OPZIONI TRA CUI        "
1210 PRINT  TAB(11);"[BIANCO][RVS]
          SCEGLIERE            "
1220 PRINT  TAB(6);"[3 DOWN][BIANC
     O][RVS] 1 [RVOFF][VERDE]  ";"
     [BIANCO][RVS] NUMERI NATURALI
      DISPOSTI "
1230 PRINT  TAB(11);"[RVS] A SPIRA
     LE            "
1240 PRINT  TAB(6);"[2 DOWN][BIANC
     O][RVS] 2 [RVOFF][VERDE]  ";"
     [BIANCO][RVS] NUMERI NATURALI
      DISPOSTI "
1250 PRINT  TAB(11);"[RVS] A TRIAN
     GOLO          "
1260 PRINT  TAB(6);"[2 DOWN][BIANC
     O][RVS] 3 [RVOFF][VERDE]  ";"
     [BIANCO][RVS] NUMERI NATURALI
      DISPOSTI "
1270 PRINT  TAB(11);"[RVS] A ESAGO
     NO            "
1280 PRINT   TAB(11);"[2 DOWN][BIA
     NCO][RVS] OPZIONE SCELTA : ";
1290 GET A:IF (A<>1 AND A<>2) AND
     A<>3 THEN 1290
1300 PRINT  TAB(16);A:FOR DE=1 TO
     2000:NEXT
1310 IF A=3 THEN 3720
1320 PRINT "[CLEAR]"; TAB(3);"[3 D
     OWN][BIANCO][RVS]RISOLUZIONE
     GRAFICA"
1330 PRINT  TAB(3);"[BIANCO][RVS]
       DELL'ELEMENTO     "
1340 PRINT "[2 DOWN][3 RIGHT][BIAN
     CO][RVS] A :   3*3 DOT "
1350 PRINT "[2 DOWN][3 RIGHT][BIAN
     CO][RVS] B :   2*2 DOT "
```

```
1360 PRINT "[2 DOWN][3 RIGHT][BIAN
     CO][RVS] C :  1*1 DOT "
1370 PRINT "[2 DOWN][3 RIGHT][BIAN
     CO][RVS]OPZIONE SCELTA ? "
1380 GET  A$:IF (A$<>"A" AND A$<>"
     B") AND A$<>"C" THEN 1380
1390 PRINT "[UP]"; TAB(20);"[BIANC
     O][RVS]";A$:FOR W=1 TO 2000:N
     EXT
1400 :
1410 REM *************************
1420 REM * ROUTINE IN LM PER LA   *
1430 REM * PULIZIA DELLO SCHERMO *
1440 REM *************************
1450 :
1460 FOR K=49152 TO 49222:READ P:P
     OKE K,P:NEXT
1470 BA=8192: SYS49152
1480 DATA  173,24,208,9,8,141,24,2
     08
1490 DATA 173,17,208,9,32,141,17,2
     08
1500 DATA  169,0,133,251,169,32,13
     3,252
1510 DATA 160,0,169,0,145,251,200,
     192
1520 DATA 0,208,249,230,252,169,64
     ,197
1530 DATA 252,208,239,169,0,133,25
     1,169
1540 DATA 4,133,252,160,0,169,3,14
     5
1550 DATA 251,200,192,0,208,249,23
     0,252
1560 DATA 169,8,197,252,208,239,96
1570 IF A$="A" THEN V=15
1580 IF A$="B" THEN V=10
1590 IF A$="C" THEN V=7
1600 IF  A=1 AND A$="A" THEN 1760
1610 IF  A=1 AND A$="B" THEN 2040
1620 IF  A=1 AND A$="C" THEN 2220
1630 IF  A=2 AND A$="A" THEN 2390
1640 IF  A=2 AND A$="B" THEN 2610
1650 IF  A=2 AND A$="C" THEN 2820
1660 IF  A=3 AND A$="A" THEN 3030
1670 IF  A=3 AND A$="B" THEN 3300
1680 IF  A=3 AND A$="C" THEN 3550
1690 :
1700 REM *************************
1710 REM * DISEGNA UNA SPIRALE    *
1720 REM * IN RISOLUZIONE         *
1730 REM *   3 X 3  DOT           *
1740 REM *************************
1750 :
1760 CL(1)=-3:CL(2)=960:CL(3)=3:CL
     (4)=-960:CL(7)=0:CL(8)=-1:CL(
     9)=-2:CL(10)=318
1770 CL(11)=638:CL(12)=639:CL(13)=
     640:CL(14)=320:CL(15)=319
1780 FOR  A1=1 TO 2
1790 U1=32097+3*(A1):TU=U1:GOSUB 1
     910
1800 NEXT A1
1810 U1=31143:F=2:E=1:T1=-1
1820 T1=T1+1:Q1=CL(T1):IF T1=4 THE
     N T1=0
1830 IF Q1=-3 OR Q1=3 THEN E=E+1
1840 FOR TU=(U1+Q1) TO (U1+(E*Q1))
      STEP Q1
1850 GOSUB  1880
1860 NEXT TU:U1=U1+E*Q1:GOTO 1820
1870 :
1880 F=F+1
1890 FOR G1=2 TO  SQR(F):IF F/G1=
     INT(F/G1) THEN  RETURN
1900 NEXT G1
1910 FOR  Z1=7 TO V
1920 Y= INT((TU+CL(Z1))/320):X=TU+
     CL(Z1)-Y*320
1930 CH= INT(X/8):RO= INT(Y/8):LN=
     Y AND 7
1940 BI=7-(X AND 7)
1950 BY=BA+RO*320+8*CH+LN
1960 POKE BY, PEEK(BY) OR (2^BI):N
     EXT Z1:RETURN
1970 :
1980 REM *************************
1990 REM * DISEGNA UNA SPIRALE    *
2000 REM * IN RISOLUZIONE         *
2010 REM *   2 X 2  DOT           *
2020 REM *************************
2030 :
2040 CL(1)=-2:CL(2)=640:CL(3)=2:CL
     (4)=-640:CL(7)=0:CL(8)=-1
2050 CL(9)=319:CL(10)=320
2060 FOR A1=1 TO 2
2070 U1=32098+(2*A1):TU=U1:GOSUB
     1910
2080 NEXT A1
2090 U1=31462:F=2:E=1:T1=-1
2100 T1=T1+1:Q1=CL(T1):IF  T1=4 TH
     EN T1=0
```

```
2110 IF  Q1=-2 OR Q1=2 THEN  E=E+1
2120 FOR TU=(U1+Q1) TO (U1+(E*Q1))
     STEP Q1
2130 GOSUB 1880
2140 NEXT TU:U1=U1+E*Q1:GOTO 2100
2150 :
2160 REM ************************
2170 REM * DISEGNA UNA SPIRALE    *
2180 REM * IN RISOLUZIONE         *
2190 REM *   1 X 1  DOT           *
2200 REM ************************
2210 :
2220 CL(1)=-1:CL(2)=320:CL(3)=1:CL
     (4)=-320
2230 FOR A1=1 TO 2
2240 U1=32099+A1:TU=U1:GOSUB 1910
2250 NEXT A1
2260 U1=31781:F=2:E=1:T1=-1
2270 T1=T1+1:Q1=CL(T1):IF T1=4 THE
     N T1=0
2280 IF Q1=-1 OR Q1=1 THEN E=E+1
2290 FOR TU=(U1+Q1) TO (U1+(E*Q1))
     STEP Q1
2300 GOSUB 1880
2310 NEXT TU:U1=U1+E*Q1:GOTO 2270
2320 :
2330 REM ************************
2340 REM * DISEGNA UN TRIANGOLO   *
2350 REM * IN RISOLUZIONE         *
2360 REM *   3 X 3  DOT           *
2370 REM ************************
2380 :
2390 CL(1)=957:CL(2)=3:CL(3)=-963:
     CL(7)=0:CL(8)=-1:CL(9)=-2:CL(
     10)=318
2400 CL(11)=638:CL(12)=639:CL(13)=
     640:CL(14)=320:CL(15)=319
2410 FOR A1=1 TO 2
2420 U1=32147+3*A1:TU=U1:GOSUB 191
     0
2430 NEXT A1
2440 E=1:F=2:U1=31190:GOTO 2510
2450 T1=T1+1:IF T1=4 THEN T1=1
2460 Q1=CL(T1)
2470 IF T1=2 THEN N=N+1
2480 IF T1=1 THEN E=E+1
2490 IF T1=2 THEN E=E+(1+2*N)
2500 IF T1=3 THEN E=E-2*N
2510 FOR TU=(U1+Q1) TO (U1+(E*Q1))
     STEP Q1
2520 GOSUB 1880
2530 NEXT TU:U1=U1+E*Q1:GOTO 2450
2540 :
2550 REM ************************
2560 REM * DISEGNA UN TRIANGOLO   *
2570 REM * IN RISOLUZIONE         *
2580 REM *   2 X 2  DOT           *
2590 REM ************************
2600 :
2610 CL(1)=638:CL(2)=2:CL(3)=-642:
     CL(7)=0:CL(8)=-1:CL(9)=319:CL
     (10)=320
2620 FOR A1=1 TO 2
2630 U1=32147+2*A1:TU=U1:GOSUB 191
     0
2640 NEXT A1
2650 U1=31509:E=1:F=2:GOTO 2720
2660 T1=T1+1:IF T1=4 THEN T1=1
2670 Q1=CL(T1)
2680 IF T1=2 THEN N=N+1
2690 IF T1=1 THEN E=E+1
2700 IF T1=2 THEN E=E+(1+2*N)
2710 IF T1=3 THEN E=E-2*N
2720 FOR TU=(U1+Q1) TO (U1+(E*Q1))
     STEP Q1
2730 GOSUB 1880
2740 NEXT TU:U1=U1+E*Q1:GOTO  2660
2750 :
2760 REM ************************
2770 REM * DISEGNA UN TRIANGOLO   *
2780 REM * IN RISOLUZIONE         *
2790 REM *   1 X 1  DOT           *
2800 REM ************************
2810 :
2820 CL(1)=319:CL(2)=1:CL(3)=-321
2830 FOR  A1=1 TO 2
2840 U1=32147+A1:TU=U1:GOSUB 1910
2850 NEXT A1
2860 U1=31828:E=1:F=2:GOTO 2930
2870 T1=T1+1:IF T1=4 THEN T1=1
2880 Q1=CL(T1)
2890 IF T1=2 THEN N=N+1
2900 IF T1=1 THEN E=E+1
2910 IF T1=2 THEN E=E+(1+2*N)
2920 IF T1=3 THEN E=E-2*N
2930 FOR TU=(U1+Q1) TO (U1+(E*Q1))
     STEP Q1
2940 GOSUB 1880
2950 NEXT TU:U1=U1+E*Q1:GOTO 2870
2960 :
2970 REM ************************
2980 REM * DISEGNA UN ESAGONO     *
```

FACIT STAMPANTI A MATRICE

4542
250 ÷ 535 cps
2 colori

4544
225 ÷ 535 cps
4 colori

MODULO CONTINUO
156 COLONNE
8 INTERFACCE ED EMULAZIONI
MACROCARATTERI IN MULTIDIMENSIONE
12 CODICI BARRE
GRAFICA PLOT 10 INTELLIGENTE
ALTO CARICO DI LAVORO (HEAVY DUTY)
ALTA AFFIDABILITA'

MILANO: Via Toffetti, 2 - Milano - Tel. 02/5694245
ANCONA: Via E. Profili, 16/18 - Fabriano - Tel. 0732/22259
BERGAMO: Via Borgo Palazzo, 90 - Bergamo - Tel. 035/233909
BOLOGNA: P.zza Porta Mascarella, 7 - Bologna - Tel. 051/238105
CAGLIARI: Via Pascoli, 3 - Decimomannu - Tel. 070/961600
CATANIA: V.le Vittorio Veneto, 160 - Catania - Tel. 095/372011
FIRENZE: Via Don Minzoni, 9 - Calenzano - Tel. 055/8879246
GENOVA: Piazzetta del Centro, 17 - Arenzano - Tel. 010/9111677
PERUGIA: Via Marzabotto, 18 - Castiglione Del Lago - Tel. 075/953844
ROMA: Via M. Garosi, 98 - Roma - Tel. 06/5204548
SALERNO: Via Raffaele Mauri, 15 - Salerno - Tel. 089/324721
TARANTO: V.le Virgilio, 53 - Taranto - Tel. 099/339916
TORINO: Via Artisti, 36 - Torino - Tel. 011/832620
TRAPANI: Via Orti, 63 - Trapani - Tel. 0923/27981
TREVISO: Via Borgo Treviso, 45 - Castelfranco - Tel. 0423/496222

FACIT
DATA
PRODUCTS

Via Toffetti, 2 - 20139 MILANO (ITALIA) - Tel. (02) 5694245 6 7 8 9

```
2990 REM * IN RISOLUZIONE          *
3000 REM *   3 X 3  DOT            *
3010 REM *************************
3020 :
3030 CL(1)=-963:CL(2)=-3:CL(3)=957
     :CL(4)=963:CL(5)=3:CL(6)=-957
3040 CL(7)=0:CL(8)=-1:CL(9)=-2:CL(
     10)=318:CL(11)=638:CL(12)=639
3050 CL(13)=640:CL(14)=320:CL(15)=
     319
3060 FOR A1=1 TO 2
3070 U1=32147+3*A1:TU=U1:GOSUB 191
     0
3080 NEXT A1:F=2
3090 FOR T3=1 TO X1-1
3100 U1=32153+3*T3:TU=U1:GOSUB  18
     80
3110 NEXT T3
3120 U1=32153+3*(T3-1)
3130 T1=T1+1:IF T1=7 THEN T1=1
3140 Q1=CL(T1)
3150 ON T1GOTO 3160,3180,3190,3160
     ,3180,3190
3160 IF T1=1 THEN E=E+1
3170 IF T1=1 OR T1=4 THEN FU=E
3180 IF T1=2 OR T1=5 THEN E=X1- SG
     N(1+(-1)^(T1))
3190 IF T1=3 OR T1=6 THEN E=FU
3200 FOR TU=(U1+Q1) TO (U1+(E*Q1))
      STEP Q1
3210 GOSUB 1880
3220 NEXT TU:U1=U1+E*Q1:GOTO 3130
3230 :
3240 REM *************************
3250 REM * DISEGNA UN ESAGONO     *
3260 REM * IN RISOLUZIONE         *
3270 REM *   2 X 2  DOT           *
3280 REM *************************
3290 :
3300 CL(1)=-642:CL(2)=-2:CL(3)=638
     :CL(4)=642:CL(5)=2:CL(6)=-638
3310 CL(7)=0:CL(8)=-1:CL(9)=319:CL
     (10)=320
3320 FOR A1=1 TO 2
3330 U1=32147+2*A1:TU=U1:GOSUB 191
     0
3340 NEXT A1:F=2
3350 FOR T3=1 TO X1-1
3360 U1=32151+2*T3:TU=U1:GOSUB 188
     0
3370 NEXT T3
3380 T1=T1+1:IF T1=7 THEN T1=1
3390 Q1=CL(T1)
3400 ON T1GOTO 3410,3430,3440,3410
     ,3430,3440
3410 IF T1=1 THEN E=E+1
3420 IF T1=1 OR T1=4 THEN FU=E
3430 IF T1=2 OR T1=5 THEN E=X1- SG
     N(1+(-1)^(T1))
3440 IF T1=3 OR T1=6 THEN E=FU
3450 FOR TU=(U1+Q1) TO (U1+(E*Q1))
      STEP Q1
3460 GOSUB 1880
3470 NEXT TU:U1=U1+E*Q1:GOTO 3380
3480 :
3490 REM *************************
3500 REM * DISEGNA UN ESAGONO     *
3510 REM * IN RISOLUZIONE         *
3520 REM *   1 X 1  DOT           *
3530 REM *************************
3540 :
3550 CL(1)=-321:CL(2)=-1:CL(3)=319
     :CL(4)=321:CL(5)=1:CL(6)=-319
3560 FOR A1=1 TO 2
3570 U1=32147+A1:TU=U1:GOSUB 1910
3580 NEXT A1:F=2
3590 FOR T3=1 TO X1-1
3600 U1=32149+T3:TU=U1:GOSUB 1880
3610 NEXT T3
3620 T1=T1+1:IF T1=7 THEN T1=1
3630 Q1=CL(T1)
3640 ON T1GOTO 3650,3670,3680,3650
     ,3670,3680
3650 IF T1=1 THEN E=E+1
3660 IF T1=1 OR T1=4 THEN FU=E
3670 IF T1=2 OR T1=5 THEN E=X1- SG
     N(1+(-1)^(T1))
3680 IF T1=3 OR T1=6 THEN E=FU
3690 FOR TU=(U1+Q1) TO (U1+E*Q1) S
     TEP Q1
3700 GOSUB 1880
3710 NEXT TU:U1=U1+E*Q1:GOTO 3620
3720 PRINT "[CLEAR]"; TAB(7);"[4 D
     OWN][BIANCO][RVS]QUALE VALORE
      DESIDERI PER "
3730 PRINT  TAB(7);"[BIANCO][RVS]
                                 "
3740 PRINT  TAB(7);"[BIANCO][RVS]L
     A BASE DELL'ESAGONO ?      ";
3750 GET  X1:IF X1=0 THEN 3750:PRI
     NT X1
3760 GOTO 1320
```

6100

DUM 1541

di **Miki i2YNJ**

Quando un possessore di Commodore 64 o Vic 20 acquista l'unità a floppy disc modello **CBM1541** o **1540**, si trova inevitabilmente a scoprire un nuovo mondo di comandi. Ciò naturalmente crea un pò di panico ed è inevitabile un pò di "casino" in testa.

Fino a ieri eravamo abituati a salvare su nastro i nostri programmi (SAVE "pippo"), oppure memorizzare dati (OPEN "dati", 1,1), senza curarci d'altro se non di inserire una cassetta vergine nel registratore e seguire i comandi suggeriti dal computer medesimo (PRESS PLAY ON TAPE).

Con i dischi, le cose sembrano più complicate. Ma non è vero. Anzi ci accorgeremo che, man mano ci familiarizzeremo con il floppy, solamente pensare di tornare al registratore ci sembrerà una cosa addirittura tediosa.

Per facilitare un primo approcio con l'unità floppy e per avere uno strumento di utilità anche in futuro, vi suggeriamo di digitare questo programma:

Il · DUM 1541

Questo programma, che permette all'utilizzatore di un home computer Commodore una veloce e semplice gestione del drive 1541 o 1540, raggruppa delle routine che servono per la formattazione e per il controllo della directory del disco.

Dopo aver digitato per intero il programma sarà sufficiente attivare le varie funzioni semplicemente inserendo il numero corrispondente alla riga del menu che appare su video e quindi premere RETURN.

Qui di seguito vengono riportare le varie spiegazioni delle funzioni del programma - DUM 1541 -, ed insieme alle spiegazio-

COMPUTER QUESTO MESE È QUESTO

COMPUTER

N.67 - lire 3000 il "newsmagazine" dell'informatica

Computer - Anno VIII - n. 67 - 25 maggio - 25 giugno 1984 - Sped. Abb. post. gr. III/70 Tassa pagata per IP - Distr. MePe

Digitare stanca

Era Ibm, anno secondo

Sempre più veloci, ma perchè?

Computer con tatto

Benchmark: Rainbow 100+ Digital Equipment

La valigia delle idee

ni anche il parallelismo dei comandi che avremmo dovuto digitare per ottenere il medesimo effetto.

1. Directory disco. Questa funzione legge la directory del disco, elencando i programmi ed i files che sono stati scritti. In modo diretto avremmo dovuto digitare:

LOAD "$", 8 (RETURN)
LIST (RETURN)

Premere RETURN per tornare al menù.

2. Formattazione nuovo disco. Il tuo computer, o meglio l'unità disco, non può utilizzare un disco nuovo che hai appena comprato in quanto questo è vergine non è quindi ancora formattato. Prima della formattazione vera e propria il programma chiede il nome che vogliamo attribuire al disco, ed anche un identificatore costituito da due caratteri alfanumerici. In modo diretto avremmo dovuto digitare:

OPEN 15,8,15
PRINT , 15,"NEWO: nome del disco, identificatore"
CLOSE 15

3. Inizializzazione disco. A volte è possibile che un identificatore sia uguale a quello di un altro dischetto e quest'ultimo sia stato il precedente nel drive, oppure può accadere che si verifichi un errore di lettura della directory del disco. In questi casi è necessario dare il comando di inizializzazione, il modo diretto avremmo dovuto digitare:

OPEN 15,8,15
PRINT ,15, "INITIALIZEO"
CLOSE15

4. Copia file su disco. Serve per copiare un file sullo stesso disco cambiando naturalmente il nome.

Se la spia rossa lampeggia si ha una condizione di errore che può essere causata da varie ragioni (nome già presente, disco pieno, etc.). Per conoscere l'eventuale ragione di errore potete usare la funzione 11. ERRORE DI STATO. Questa non copia i files protetti.

In modo diretto avremmo dovuto digitare:

OPEN 15,8,15
PRINT , 15, "COPYO: nome del file nuovo =0: nome del file vecchio"
CLOSE15

5. Copy file su un disco nuovo già formattato. Questa funzione legge il file dal disco sorgente e lo trasferisce nella memoria del computer. Il file a sua volta viene trasferito dalla RAM al disco destinazione.

La massima capacità di copia di questa opzione è di 66 blocchi. Questa routine non è molto veloce poichè impiega un minuto per sei blocchi di programma.

In modo diretto avremmo dovuto digitare:
LOAD "nome del programma", 8
cambio del disco
SAVE "nome del programma", 8

6. Copia di programmi ***DOS***. Questa opzione serve per poter trasferire velocemente i programmi DOS dal disco test all'interno del buffer e da questo verso un altro disco nuovo e formattato. In modo diretto avremmo dovuto digitare per ogni programma:

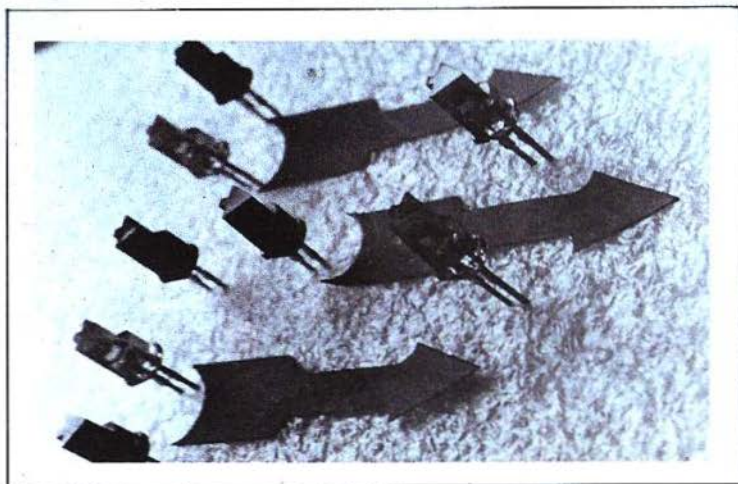
LOAD " nome del programma", 8,1
cambio del disco
SAVE "nome del programma", 8,1

7. Rinomina file. Serve per sostituire in nome di un file con il nuovo nome.

In modo diretto avremmo dovuto digitare:

OPEN 15,8,15
PRINT , 15, "RENAMEO: nome del file nuovo = nome del file vecchio"
CLOSE15

8. Cancella file. Per cancellare uno o più files errati o indesiderati dal disco basta digitare il nome del file da cancellare.

In modo diretto avremmo dovuto

digitare:

OPEN 15,8,15
PRINT 15, "SCRATCHO: nome del file"
CLOSE15

9. Convalida file. Si utilizza questa funzione per riordinare la directory del disco, chiude in oltre i files rimasti aperti e cancella quelli non correttamente cancellati.

In modo diretto avremmo dovuto digitare:

OPEN 15,8,15
PRINT 15, "VALIDCATEO"
CLOSE15

10. Copia il gesti disco. Serve per salvare il DUM 1541 su un dischetto qualsiasi senza intervenire con il load e il save.

Il modo diretto avremmo dovuto digitare:

SAVE "nome del programma"

11. Errore di stato. Questa routine controlla e informa l'operatore su eventuali errori del drive visualizzando il tipo errore. Dal numero d'errore poi si può risalire alla descrizione dell'errore consultando il libretto d'istruzione del drive commodore.

In modo diretto avremmo dovuto digitare:

OPEN 15,8,15
INPUT 15, A$, B$, C$, D$
PRINT A$, B$, C$, D$,
CLOSE15

12. Ritorno al Basic. Questa è la fine del programma serve per ritornare ad usare il calcolatore normalmente. Il programma DUM 1541 rimmarrà comunque in memoria.

```
  3 POKE 49152,10:IF  PEEK(49152)
    <>10 THEN CO=1:GOTO 6
  5 CO=2
  6 DIM  DOS%(65)
  7 FR= FRE(0):IF FR<0 THEN FR=FR
    +65536
  8 S=(FR-400)/2:M2= INT(S/256)+1
  9 DIM  TEMP%(S)
 10 PRINT "[CLEAR]";
 11 REM *****************
 12 REM **  I2YNJ MIKE   **
 13 REM **    MONZA MI   **
 14 REM *****************
 15 PRINT "  *****************"
 20 PRINT "  *  GESTI  DISCO  *"
 25 PRINT "  *****************"
 50 PRINT :PRINT " 1.DIRECTORY DI
    SCO"
 60 PRINT " 2.FORMATTAZIONE NUOVO
     DISCO"
 70 PRINT " 3.INIZIALIZZAZIONE DI
    SCO"
 80 PRINT " 4.COPIA FILE SU DISCO
    "
 85 PRINT " 5.COPIA FILE SU NUOVO
    (FORMATTATO)DISCO
 88 PRINT " 6.COPIA ENTRAMBI DOS
    WEDGE PROGRAMMI"
 90 PRINT " 7.RINOMINA FILE"
100 PRINT " 8.CANCELLA FILE(S)"
110 PRINT " 9.CONVALIDA FILES"
120 PRINT "10.COPIA IL GESTI DISC
    O"
130 PRINT "11.ERRORE STATO "
140 PRINT "12.RITORNO AL BASIC":P
    RINT
170 INPUT "SCELTA     [4 LEFT]";CH
    OICE
180 IF (CHOICE<1) OR (CHOICE>12)
    THEN PRINT "[UP]";:GOTO 170
200 ON  CHOICE GOSUB 250,300,350,
    400,800,1200,450,500,550,600,
    650,700
210 GOTO 10
250 REM  *** DISPLAY DIRECTORY **
    *
251 PRINT "[CLEAR]"
252 OPEN 1,8,0,"$"
253 GET #1,A$,B$
254 GET #1,A$,B$
256 GET #1,A$,B$
258 C=0:IF A$<>"" THEN C= ASC(A$)
260 IF B$<>"" THEN C=C+ ASC(B$)*2
    56
262 PRINT  MID$( STR$(C),2); TAB(
    3);
264 GET #1,B$:IF ST<>0 THEN 282
266 IF B$<> CHR$(34) THEN 264
268 GET #1,B$:IF B$<> CHR$(34) TH
    EN PRINT B$;:GOTO 268
270 GET #1,B$:IF B$= CHR$(32) THE
    N 270
272 PRINT  TAB(18);:C$=""
274 C$=C$+B$:GET #1,B$:IF B$<>""
    THEN 274
276 PRINT  LEFT$(C$,3)
280 IF ST=0 THEN 254
282 PRINT "BLOCCHI LIBERI"
```

```
284 CLOSE 1:PRINT :PRINT :PRINT "
    PREMI UN TASTO QUALUNQUE PER
    IL MENU'"
290 GET X$:IF X$="" THEN 290
295 RETURN
300 REM  *** FORMATTAZIONE DISCO
    ***
301 PRINT "[CLEAR]"
305 PRINT "INSERISCI IL DISCO":PR
    INT "DA FORMATTARE"
310 PRINT "TITOLO DEL DISCO ":INP
    UT  DISK$
320 PRINT :PRINT "IDENTIFICATORE
    DISCO  ":INPUT  EXT$
325 MACRO$="NO:"+DISK$+","+EXT$
330 OPEN 15,8,15,MACRO$
340 CLOSE  15:MACRO$="":RETURN
350 REM  *** INIZIALIZAZIONE DISC
    O"
355 PRINT "[CLEAR]"
360 PRINT "INSERISCI IL DISCO":PR
    INT "DA INIZIALIZZARE":PRINT
370 PRINT "PREMI(RETURN) PER ":PR
    INT "INIZIALIZZARE":INPUT X$
380 OPEN 15,8,15,"I"
390 CLOSE 15:RETURN
400 REM  *** COPIA DEL FILE SULLO
     STESSO DISCO ***
405 PRINT "[CLEAR]":PRINT :PRINT
410 PRINT "NOME DEL FILE INIZIALE
    ":INPUT DISK$
420 PRINT :PRINT "NUOVO NOME DEL
    FILE ":INPUT NWS$
425 MACRO$="C:"+NWS$+"="+DISK$
430 OPEN 15,8,15,MACRO$
440 CLOSE 15:MACRO$="":RETURN
450 REM  *** RINOMINA FILE ***
455 PRINT "[CLEAR]":PRINT
460 PRINT "VECCHIO NOME DEL FILE"
    :INPUT  DISK$
470 PRINT :PRINT "NUOVO NOME DEL
    FILE ":INPUT  NWS$
475 MACRO$="R:"+NWS$+"="+DISK$
480 OPEN 15,8,15,MACRO$
490 CLOSE 15:MACRO$="":RETURN
500 REM  *** CANCELLAZIONE FILE *
    **
505 PRINT "[CLEAR]":PRINT :PRINT
510 PRINT "NOME DEL FILE DA CANCE
    LLARE":INPUT  DISK$
520 PRINT :PRINT :PRINT "PREMI QU
    ALUNQUE TASTO PER CANCELLARE"
525 GET X$:IF X$="" THEN 525
530 MACRO$="SO:"+DISK$
535 OPEN 15,8,15,MACRO$
540 CLOSE 15:MACRO$="":RETURN
550 REM  *** CONVALIDA FILES ***
555 PRINT "[CLEAR]":PRINT :PRINT
560 PRINT "PERICOLO : FILES APERT
    O":PRINT "POTREBBE ESSERE CAN
    CELLATO"
570 PRINT :PRINT :PRINT "PREMI QU
    ALUNQUE TASTO PER CONVALIDARL
    O"
575 GET X$:IF X$="" THEN 575
580 OPEN 1,8,15,"V"
590 CLOSE 1:RETURN
600 REM *** SALVA IL GESTI DISCO
    ***
605 PRINT "[CLEAR]":PRINT :PRINT
610 PRINT "INSERISCI IL DISCO DA
    SCRIVERE":PRINT
620 PRINT :PRINT :PRINT "PREMI QU
    ALUNQUE TASTO PER SALVARE IL
    GESTI DISCO"
622 GET X$:IF X$="" THEN 622
625 OPEN 1,8,15
630 SAVE "@O:GESTI DISCO",8
635 CLOSE 1
640 RETURN
650 REM  *** VISUALIZZA LA CONDIZ
    IONE DELL'ERRORE ***
655 OPEN 15,8,15
660 INPUT#15,A$,B$,C$,D$
670 PRINT "[CLEAR]":PRINT :PRINT
680 PRINT "CONDIZIONE D' ERRORE :
     "B$:PRINT :PRINT "ERRORE NU
    MERO:  "A$
690 PRINT :PRINT :PRINT "PREMI QU
    ALUNQUE TASTO PER RITORNARE"
695 GET X$:IF X$="" THEN 695
697 CLOSE 15:RETURN
700 REM  *** RITORNO AL BASIC ***
705 PRINT "[CLEAR]":PRINT :PRINT
710 PRINT "NOTA:IL PROGRAMMA GEST
    I DISCO E'ANCORA MEMORIZZATO"
720 END
800 REM  *** COPIA DEL FILE ***
801 PRINT "[CLEAR]"
802 CLOSE 15
803 MAX= INT(S/256)
804 PRINT "LA DIMENSIONE DEL FILE
```

```
     CHE PUO'ESSERE    COPIATO E'
     "MAX"BLOCCHI":PRINT
 806 IF MAX>=M2 THEN GOTO 809
 807 PRINT "PER COPIARE FILE SUPER
     IORI A"M2"BLOCCHI,";
 808 PRINT "SPEGNI E RIACCENDI IL
     COMPUTER RICARICANDO IL PRG"
 809 PRINT :PRINT "  LEGGE/SALVA C
     IRCA 6 BLOCCHI AL MINUTO"
 810 PRINT :PRINT "SCRIVI IL NOME
     DEL FILE DA COPIARE"
 820 INPUT  FILE$
 830 PRINT :PRINT "   INPUT FILE T
     YPE:"
 840 PRINT "      P PROGRAMMA"
 850 PRINT "      S SEQUENZIALE"
 860 PRINT "      U UTENZE"
 870 PRINT "      R RELATIVO"
 880 INPUT  TYPE$
 890 PRINT :PRINT "INSERIRE IL DIS
     CO SORGENTE E PIGIARE.... <RET
     URN>"
 900 GET W$:IF W$="" THEN GOTO 900
 910 GOSUB 1000:REM  READ FILE INT
     O TEMP
 920 PRINT :PRINT "INSERIRE IL DIS
     CO DESTINAZIONE E PIGIARE. <RE
     TURN>"
 930 GET W$:IF W$="" THEN GOTO 930
 940 GOSUB 1050:REM  WRITE FILE FR
     OM TEMP$
 950 PRINT :PRINT "ESEGUI UN ALTRA
      COPIA ";:INPUT W$
 960 IF  LEFT$(W$,1)="Y" THEN GOTO
      920
 970 GOTO 1100
 980 REM  - TEST PER UNA BUONA APE
     RTURA
 985 INPUT#15,A$,B$,C$,D$
 990 IF  VAL(A$)=0 THEN RETURN
 995 PRINT A$,B$,C$,D$:CLOSE 15:ST
     OP
1000 REM  - READ FILE INTO TEMP%
1002 I=1
1004 OPEN 15,8,15
1006 OPEN 5,8,5,"O:"+FILE$+","+TYP
     E$+",R"
1008 GOSUB 980
1009 PRINT "       STO LEGGENDO. . .
     "
1010 GET #5,A$
1012 TEMP%(I)= ASC(A$+ CHR$(0))
1014 I=I+1
1016 IF ST=0 THEN 1010
1018 PRINT :PRINT  "FILE=? ";FILE$
     ,"ST=";ST,"BYTES=";I:CLOSE  5
     :CLOSE  15
1020 RETURN
1050 REM  - WRITE FILE FROM TEMP%
1052 OPEN 15,8,15
1054 OPEN 5,8,5,"O:"+FILE$+","+TYP
     E$+",W"
1056 GOSUB 980
1057 PRINT "       STO SALVANDO. . .
     "
1058 J=1
1060 PRINT#5, CHR$(TEMP%(J));
1062 J=J+1
1064 IF J<I THEN 1060
1066 PRINT :PRINT "FILE= ";FILE$,"
     BYTES=";J:CLOSE 5:CLOSE 15
1068 RETURN
1100 RETURN
1200 REM  *** COPIA DEI PROGRAMMI
     DOS ***
1205 PRINT "[CLEAR]":PRINT "INSERI
     RE IL DISCO SORGENTE CON  WED
     GE  E  DOS"
1210 PRINT " PROGRAMM(I) E PIGIA <
     RETURN>"
1215 GET W$:IF W$="" THEN GOTO 121
     5
1218 TYPE$="P"
1219 IF CO=1 THEN FILE$="VIC-20 WE
     DGE":GOTO 1235
1220 FILE$="C-64 WEDGE":GOSUB 1000
1225 FOR C=1 TO 64:DOS%(C)=TEMP%(C
     ):NEXT  C:REM  MOVE C-64 WEDG
     E TO SMALL BUFFER
1230 FILE$="DOS 5.1"
1235 GOSUB 1000:PRINT :PRINT "INSE
     RIRE IL DISCO DESTINAZIONE E
     PIGIA <RETURN>"
1240 GET W$:IF W$="" THEN GOTO 124
     0
1245 GOSUB 1050:IF CO=1 THEN RETUR
     N
1250 FOR C=1 TO 64:TEMP%(C)=DOS%(C
     ):NEXT  C
1255 FILE$="C-64 WEDGE":I=64:GOSUB
      1050:REM  WRITE C-64 WEDGE
1290 RETURN
```

Se vuoi abbonarti

Registrate il mio abbonamento annuale a Commodore.

☐ Ho versato oggi stesso il canone di Lire 20.000 a mezzo c/c postale n o 30426209 intestato a: Sinclair Computer - V.le Famagosta, 75 - 20142 Milano

☐ Acchiudo assegno per lire 20.000 banca .... n o .... a favore di Sinclair Computer

Il mio computer è: uno ZX81 ☐ uno Spectrum ☐ altro (specificare) ....

Ho ☐ / non ho ☐ la stampante, ma voglio ☐ comprarla.

Preferisco programmi di gioco ☐, didattici ☐, d'utilità ☐, altro ....

Nome ....

Via .... nº .... CAP. ☐☐☐☐☐ Città ....

Tel. .... Orario ....

Se vuoi collaborare

Registratemi fra i collaboratori regolari di Commodore.

A titolo di prova vi invio un articolo e la cassetta col programma " .... " di cui vi garantisco l'assoluta originalità autorizzandovene la pubblicazione.

Per il compenso ☐ telefonatemi al .... ☐ Scrivetemi all'indirizzo sottoindicato.

Nome ....

Via .... nº .... CAP. ☐☐☐☐☐ Città ....

Tel. .... Orario ....

Se vuoi un consiglio o consigliarci

HELP ....

Nome ....

Via .... nº .... CAP. ☐☐☐☐☐ Città ....

Tel. .... Orario ....

Se vuoi votare il software

Questo mese ho comprato i seguenti programmi e li ho giudicati così:

Titolo	Giudizio d'utente (mettere una x nella casella voluta)			
	Ottimo	Buono	Mediocre	Deludente

Nome ....

Via .... nº .... CAP. ☐☐☐☐☐ Città ....

Tel. .... Orario ....
Facoltativo

Se vuoi vendere o comprare

Vendo ☐ Compro ☐ ....

Nome ....

Via .... nº .... CAP. ☐☐☐☐☐ Città ....

Tel. .... Orario ....

Da inviare in busta chiusa a:

Spett.le rivista
Commodore
Systems Editoriale

v.le Famagosta, 75
20142 Milano

Si, voglio abbonarmi

Da inviare in busta chiusa a:

Spett.le rivista
Commodore
Systems Editoriale

v.le Famagosta, 75
20142 Milano

Si, voglio collaborare

Da inviare in busta chiusa a:

Spett.le rivista
Commodore
Systems Editoriale

v.le Famagosta, 75
20142 Milano

Si, chiedo consiglio

Da inviare in busta chiusa a:

Spett.le rivista
Commodore
Systems Editoriale

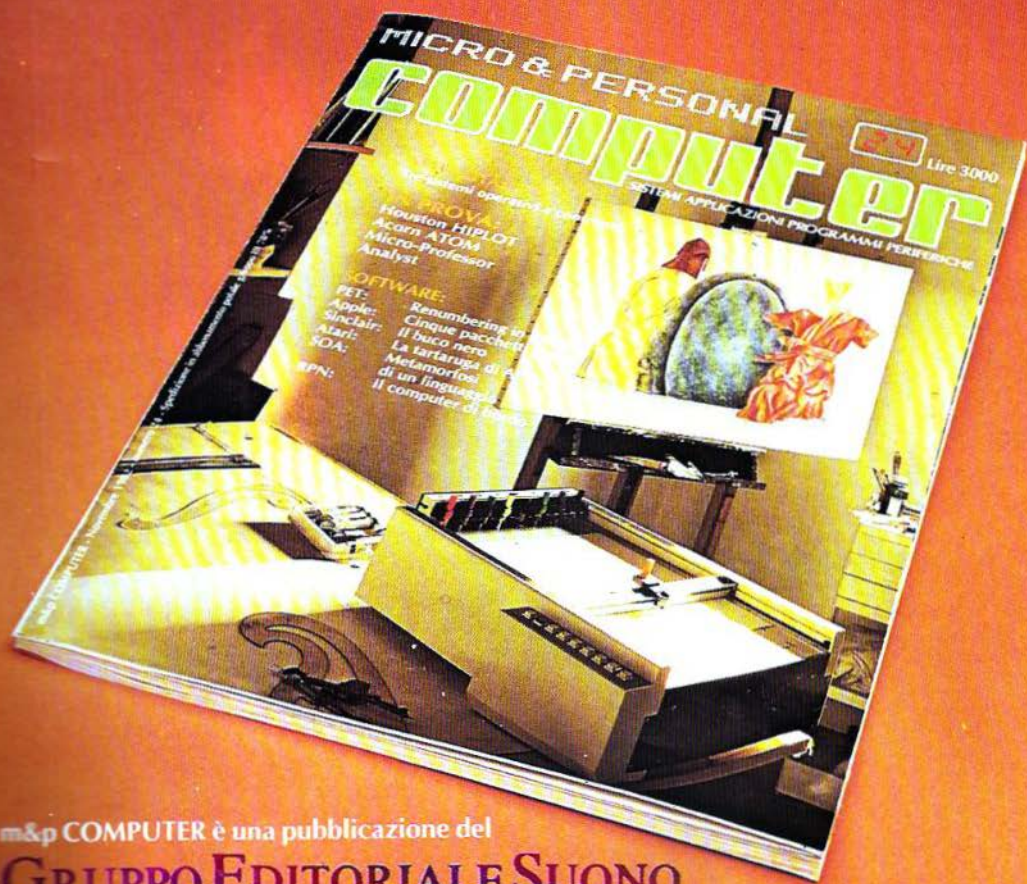
v.le Famagosta, 75
20142 Milano

Si, voglio votare

Da inviare in busta chiusa a:

Spett.le rivista
Commodore
Systems Editoriale

v.le Famagosta, 75
20142 Milano

Si vendo/compro

FINALMENTE.
LA TAVOLETTA GRAFICA A PIENE PRESTAZIONI AD UN PREZZO ACCESSIBILE A TUTTI

Koala
Disponibile per Apple][+ e //e
Atari 400 e 800, Commodore 64
ed IBM P.C.

La tavoletta grafica KOALA è la più simpatica innovazione nel campo dei personal computers. Con KOALA, controllate il vostro computer con un dito. Più veloce di un paddle, più versatile di un joystick e più semplice di una tastiera.
La tavoletta grafica KOALA è compatibile con la maggior parte di software esistente e viene fornita completa del suo programma grafico "Micro Illustrator". KOALA-PAD è il miglior modo per creare immagini ad alta risoluzione con il vostro computer.

COMPUTER GRAPHICS DIVISION
MILANO: Via L. da Vinci, 43 - 20090 Trezzano S/N
Tel. 02/4455741/2/3/4/5 - Tlx: TELINT I 312827
ROMA: Via Salaria, 1319 - 00138 Roma
Tel. 06/6917058-6919312 - Tlx: TINTRO I 614381